

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA:  
INGENIERÍA DE SISTEMAS**

**Tesis previa a la obtención del título de: INGENIERO DE SISTEMAS**

**TEMA:  
DESARROLLO DE UN PROTOTIPO DE VIDEOJUEGO QUE PERMITA  
APOYAR LOS PROCESOS DE ENSEÑANZA APRENDIZAJE DE LOS  
HECHOS HISTÓRICOS MÁS RELEVANTES DEL ECUADOR**

**AUTORES:  
MARCO ANTONIO ALVAREZ MOREIRA  
GABRIEL ALEJANDRO GARZÓN CAJAS**

**DIRECTOR:  
FRANKLIN EDMUNDO HURTADO LARREA**

**Quito, junio del 2014**

## **DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO DEL TRABAJO DE TITULACIÓN**

Nosotros, autorizamos a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaramos que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Quito, junio del 2014

---

Marco Antonio Alvarez Moreira

CC: 1719239731

---

Gabriel Alejandro Garzón Cajas

CC: 1718496910

## **DEDICATORIA**

A mi familia, en especial a mis padres, Carmen y Marco, por siempre brindarme ese apoyo y paciencia en la realización de este trabajo.

Marco Alvarez Moreira

A quienes comparten mis alegrías y triunfos:

A mis padres, por su sacrificio, entrega y el aguante que han tenido conmigo; a mi hermano y abuelitos por creer en mí, a Sari, por acompañarme en mis alegrías, tristezas y por el apoyo incondicional.

Gabriel Garzón Cajas

## **AGRADECIMIENTO**

Al Ing. Franklin Hurtado por su apoyo, confianza y porque nos ha brindado su dedicación al realizar este proyecto

Marco Alvarez Moreira

Gabriel Garzón Cajas

## INDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1.....</b>	<b>3</b>
<b>PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>3</b>
1.1 Antecedentes .....	3
1.2 Formulación del problema .....	5
1.3 Justificación.....	5
1.4 Objetivos .....	6
1.4.1 Objetivo general .....	6
1.4.2 Objetivos específicos. ....	6
1.5 Metodología .....	8
1.5.1 SCRUM.....	8
1.5.2 Metodología SCRUM para creación de videojuegos.....	9
1.5.2.1 Roles.....	10
1.5.2.2 Ciclo de vida .....	10
1.6 Alcance.....	12
1.6.1 Aspectos PBS .....	13
1.6.1.1 Pantalla principal del juego.....	14
1.6.1.2 Exploración .....	14
1.6.1.3 Desafíos / juegos .....	14
1.6.1.4 Herramientas del jugador .....	15
1.6.1.5 Personajes jugables .....	15
1.6.2 Cambios realizados en la PBS .....	15

<b>CAPÍTULO 2.....</b>	<b>17</b>
<b>MARCO TEÓRICO .....</b>	<b>17</b>
2.1 Concepto de prototipo en ingeniería .....	17
2.2 Concepto de videojuego .....	17
2.3 Clasificación de videojuego .....	18
2.4 Desarrollo y elaboración de videojuegos .....	18
2.5 El guión .....	20
2.6 Investigación y análisis.....	21
2.6.1 Unity.....	22
2.6.1 Maya.....	23
<b>CAPÍTULO 3.....</b>	<b>24</b>
<b>DISEÑO DEL VIDEOJUEGO .....</b>	<b>24</b>
3.1 Creación del guión .....	24
3.1.1 Datos importantes sobre el guión.....	25
3.2 Definición de secuencias.....	27
3.3 Creación de personajes y ambientes .....	28
3.3.1 Personaje principal o protagonista .....	29
3.3.2 Personaje secundario .....	30
3.4 Creación del storyboard .....	30
<b>CAPÍTULO 4.....</b>	<b>34</b>
<b>DESARROLLO DEL VIDEOJUEGO.....</b>	<b>34</b>
4.1 Geometría 3d.....	34

4.2 Modelado .....	36
4.3 Texturizado de personajes y ambientes.....	38
4.4 Audio.....	44
4.5 Diseño de la interfaz.....	45
4.5.1 Exportación de los modelos de Maya a Unity .....	45
4.5.2 Exportación del modelo 3D a Unity .....	46
4.5.3 Construcción de la funcionalidad del juego .....	51
4.5.3.1 Estándares de codificación.....	51
4.5.3.1.1 Declaración de variables .....	51
4.5.3.1.2 Declaración de funciones .....	52
4.5.3.2 Script ControlSonidoPisadas.js .....	53
4.5.3.3 Script FuncionalidadCuevaScript.js.....	54
4.5.3.4 Script FuncionalidadHerramientaScript.js .....	71
4.5.3.5 Script InventarioAnimacion.js .....	78
4.5.3.6 Script InventarioScript.js .....	80
4.5.3.7 Script MensajesAyudaJuegoIglesiaScript.js .....	83
4.5.3.8 Script OpenSceneButtonScript.js.....	84
4.5.3.9 Script PantallaFinalScript.js.....	86
4.5.3.10 Script SantoDomingoActivarMecanismo.js.....	87
4.5.3.11 Script SantoDomingoControlTriggers.js .....	88
4.5.2.12 Script SantoDomingoGUITextScript.js .....	90
4.5.4 Construcción del videojuego.....	91
4.5.4.1 Script FuncionalidadLibro.cs .....	102
4.5.4.2 Script FuncionalidadMaestro.js .....	106

<b>CAPÍTULO 5.....</b>	<b>111</b>
<b>PRUEBAS Y RESULTADOS.....</b>	<b>111</b>
5.1 Pruebas de funcionalidades .....	113
5.2 Pruebas de sistema (Sistema Operativo) .....	120
5.3 Pruebas Alfa y Beta.....	123
5.3.1 Pruebas Alfa .....	123
5.3.2 Pruebas Beta.....	125
<b>CONCLUSIONES .....</b>	<b>130</b>
<b>RECOMENDACIONES.....</b>	<b>133</b>
<b>LISTA DE REFERENCIAS .....</b>	<b>135</b>



## ÍNDICE DE FIGURAS

• Figura 1. Fases del proceso .....	11
• Figura 2. Estructura de desglose del producto .....	12
• Figura 3. Imagen de posible protagonista .....	29
• Figura 4. Imagen protagonista secundario .....	30
• Figura 5. Imagen bocetos storyboard 1 .....	31
• Figura 6. Imagen bocetos storyboard 2 .....	31
• Figura 7. Imagen bocetos storyboard 3 .....	32
• Figura 8. Imagen bocetos storyboard 4 .....	32
• Figura 9. Imagen bocetos storyboard 5 .....	33
• Figura 10. Geometría 3D de la iglesia .....	34
• Figura 11. Geometría 3D de la cámara secreta, vista lateral y de frente .....	35
• Figura 12. Geometría 3D del templo .....	35
• Figura 13. Modelado de la iglesia .....	36
• Figura 14. Modelado del templo .....	36
• Figura 15. Modelado de la cámara secreta .....	37
• Figura 16. Modelado interna de la cámara secreta .....	37
• Figura 17. Texturizado iglesia .....	38
• Figura 18. Texturizado interno iglesia .....	38
• Figura 19. Texturizado interno y vista desde arriba iglesia .....	39
• Figura 20. Texturizado interno y columnas que se levantan iglesia .....	39
• Figura 21. Texturizado interno cámara secreta, sol inca.....	40
• Figura 22. Texturizado interno cámara secreta .....	40
• Figura 23. Texturizado interno templo.....	41

• Figura 24. texturizado interno templo e iluminación .....	41
• Figura 25. Texturizado exterior montaña.....	42
• Figura 26. Texturizado exterior montaña y templo juntos.....	42
• Figura 27. Texturizado exterior montaña y templo, vista frontal .....	43
• Figura 28. Exportación modelo templo a Unity .....	47
• Figura 29. Remoción de los materiales que no son usados, hypershade. Window –RenderingEditors-Hypershade .....	47
Figura 30. remoción de los materiales que no son usados, Edit- DeleteUnusedNodes.....	48
• Figura 31. Imagen de ventana para exportar File->ExportAll .....	48
• Figura 32. Ventana de selección “fbx export” .....	49
• Figura 33. Ventana de selección de valores para exportar texturas .....	49
• Figura 34. Ventana de selección de valores recomendados por unity .....	50
• Figura 35. Carpeta sonido en unity .....	53
• Figura 36. Ejemplo de mensajes d ayuda que tiene el jugador .....	56
• Figura 37. Elemento del juego “Historia del Inca” .....	57
• Figura 38. Componente Transform .....	58
• Figura 39. Mesh del elemento Piedra Juego 1 .....	58
• Figura 40. Componente Mesh Renderer .....	59
• Figura 41. Componente Mesh Collider.....	59
• Figura 42. Material Bumped Specular con sus texturas.....	60
• Figura 43. Relación padre-hijo de dos elementos. ....	60
• Figura 44. Escena “Scene-CuevaJuego” .....	63
• Figura 45. Elementos del juego “Historia del Primer Inca”.....	63
• Figura 46. Elemento “piedrajuego1”. ....	64

• Figura 47. Consola de Unity .....	67
• Figura 48. Imagen de ayuda para el elemento “piedrajuego1”. .....	68
• Figura 49. Mensaje de error en la secuencia seleccionada.....	71
• Figura 50. Imagen del Dios Inti después de la animación al completar el mini juego “Historia del primer Inca” .....	71
• Figura 51. Mini juego en el interior de la iglesia de Santo Domingo. ....	72
• Figura 52. Nombres de los elementos que participan en el mini juego dentro de la iglesia de Santo Domingo.....	74
• Figura 53. Representación de variables booleanas públicas en la interfaz de Unity.....	75
• Figura 54. Estado de las variables booleanas públicas en la escena “scene- caminotemplo2”. .....	76
• Figura 55. Estado de las variables booleanas públicas en la escena “scene- final”.....	76
• Figura 56. Animación del inventario .....	78
• Figura 57. Inventario normal.....	82
• Figura 58. Efecto de ítem del inventario al posicionar el cursor del mouse sobre uno de ellos.....	82
• Figura 59. Pantalla del título del juego.....	84
• Figura 60. Botón Historia.....	85
• Figura 61. Botón Objetivo... ..	86
• Figura 62. Pantalla final del juego.....	87
• Figura 63. Trigger en Unity .....	88
• Figura 64. Acceso a la ventana Build Settings.....	91
• Figura 65. Orden de las escenas del juego y plataforma final para el mismo....	92

• Figura 66. Botón play en la interfaz de Unity .....	92
• Figura 67. Pantalla de inicio del juego.....	93
• Figura 68. Botones de pantalla de título.....	93
• Figura 69. Video Introductorio .....	94
• Figura 70. Mecanismo que acciona el juego “El Ciclo Solar” .....	94
• Figura 71. Pilares del mini juego “El Ciclo Solar” .....	95
• Figura 72. Selección de un pilar.....	95
• Figura 73. Ayuda para el primer mini juego .....	96
• Figura 74. Mini juego “el ciclo solar” ha sido resuelto .....	96
• Figura 75. Inicio de escena.....	97
• Figura 76. Elementos del mini juego “la historia del primer inca”.....	97
• Figura 77. Diferentes ayudas relacionadas a las imágenes en frente del jugador.....	98
• Figura 78. Resolución del mini juego “La Historia del Primer Inca”.....	98
• Figura 79. Ayuda para buscar el templo perdido .....	99
• Figura 80. Templo de Atahualpa.....	99
• Figura 81. Interior templo .....	100
• Figura 82. Selección del Dios Inca .....	100
• Figura 83. Escena final.....	101
• Figura 84. Pantalla final .....	101
• Figura 85. Inventariolibro en Unity .....	105
• Figura 86. Libro de notas .....	106
• Figura 87. Imágenes del profesor al comenzar el nivel de la iglesia de Santo Domingo.....	110

- Figura 88. Imágenes del profesor una vez activado el mini-juego de la iglesia de Santo Domingo..... 110

## ÍNDICE DE TABLAS

• Tabla 1. Prueba de funcionalidad No. 1 .....	113
• Tabla 2. Prueba de funcionalidad No. 2 .....	114
• Tabla 3. Prueba de funcionalidad No. 3 .....	114
• Tabla 4. Prueba de funcionalidad No. 4 .....	115
• Tabla 5. Prueba de funcionalidad No. 5 .....	115
• Tabla 6. Prueba de funcionalidad No. 6 .....	116
• Tabla 7. Prueba de funcionalidad No. 7 .....	116
• Tabla 8. Prueba de funcionalidad No. 8 .....	117
• Tabla 9. Prueba de funcionalidad No. 9 .....	117
• Tabla 10. Prueba de funcionalidad No. 10 .....	118
• Tabla 11. Prueba de funcionalidad No. 11 .....	118
• Tabla 12. Prueba de funcionalidad No. 12 .....	119
• Tabla 13. Prueba de funcionalidad No. 13 .....	119
• Tabla 14. Resultados pruebas de Sistema Operativo .....	121
• Tabla 15. Resultados pruebas de Sistema Operativo .....	122
• Tabla 16. Características PC .....	124
• Tabla 17. Características resolución de pantalla .....	124
• Tabla 18. Datos del jugador .....	124
• Tabla 19. Características pc prueba Beta No. 1 .....	126
• Tabla 20. Resolución pantalla, prueba Beta No. 1 .....	126
• Tabla 21. Datos jugador prueba Beta No. 1 .....	126
• Tabla 22. Apreciación prueba Beta No. 1 .....	127
• Tabla 23. Características pc prueba Beta No. 2 .....	127

- Tabla 24. Resolución pantalla, prueba Beta No. 2..... 127
- Tabla 25. Datos jugador prueba Beta No. 2 ..... 128
- Tabla 26. Apreciación prueba Beta No. 2..... 128
- Tabla 27. Características pc prueba Beta No. 3 ..... 128
- Tabla 28. Resolución pantalla, prueba Beta No. 3..... 129
- Tabla 29. Datos jugador prueba Beta No. 3 ..... 129
- Tabla 30. Apreciación prueba Beta No. 3 ..... 129

## ÍNDICE DE ANEXOS

- Anexo 1. Guión del juego ..... 138
- Anexo 2. Focus Group ..... 167



## **RESUMEN**

Este proyecto pretende ser una guía para realizar un videojuego, dar pautas para el desarrollo y ejecución de varios procesos y modelos, como solventar de buena manera ciertos inconvenientes que se puede presentar en el transcurso de la creación del videojuego. Esa así como en el primer capítulo se da una idea de como empezar el videojuego teniendo claro lo que se quiere obtener y a donde se quiere llegar. En el segundo capítulo se da unos conceptos básicos necesarios para conocer más sobre los videojuegos, en el tercer capítulo se da a conocer la importancia que tiene el guión, así como sus secuencias, personajes y ambientes; y en el cuarto capítulo se especifica en que programas (Unity, Maya) se crea el videojuego, como se logro implementarlo y que pasos son necesarios realizar para que el juego al final quede funcional, para esto se debe tener en cuenta los modelos a utilizar así como las texturas y funcionalidades necesarias para que el videojuego sea entretenido y a la vez educativo.

## **ABSTRACT**

This project is intended as a guide for a video game, provide guidelines for the development and execution of various processes and models, as certain good way to solve problems that may occur during the creation of the game. That as well as the first chapter gives an idea of how to start the game having clear what you want and get where you want to go. In the second chapter some basic concepts needed to know more about video games, in the third chapter discloses the importance of the screenplay , and their sequences , characters and environments is given ; and in the fourth chapter specifies which programs (Unity , Maya) the game is created as achievement and implement steps are required to perform the play at the end remains functional , for this must take into account the models to use and textures and functionality needed for the game is entertaining yet educational.

## INTRODUCCIÓN

La mayoría de personas en algún momento han jugado o al menos escuchado de algún tipo de videojuego, este tipo especial de software ha experimentado un avance increíble en los últimos años, tal es así que hoy en día se los puede encontrar en versiones para prácticamente todos los dispositivos y/o consolas<sup>1</sup> disponibles en el mercado.

Un videojuego es un programa en el que existe interacción entre el dispositivo y el usuario. Pueden ser de distintos tipos: consola, móvil, plataforma, ordenador, etc. El usuario en la mayoría de videojuegos controla la acción del personaje que se desarrolla en un ambiente de realidad virtual la cual se basa en ciertos objetivos que se deben alcanzar y normas que se deben cumplir para terminar el juego.

En la actualidad existen desarrollados varios modelos de dispositivos, sistemas operativos y versiones de consolas para disfrutar de los videojuegos, las empresas que se dedican exclusivamente a esto han tenido un avance importantísimo desde su aparición en los años 70. El usuario tiene varias modalidades de juego, ya que puede hacerlo por turnos o jugar simultáneamente con varias personas; esto con una amplia diversidad de temáticas y géneros.

Los niños y jóvenes han convertido a los videojuegos en una actividad común, lo que ha llevado a generar varias críticas y reflexiones, debido a los diferentes contenidos que estos presentan o al tiempo que se invierte jugándolos. Los videojuegos sin embargo dan muestras de cómo el aprendizaje y el juego se juntan en lo que se ha denominado edutenimiento (De Kerckhove, 1999), un aprendizaje que se diferencia en buena parte de aquel que se considera formal que normalmente se lo recibe en la escuela, en que los jóvenes adquieren habilidades manuales y de coordinación; en algunas situaciones es necesario el uso estrategias como la toma de decisiones y resolución de problemas, es así

---

<sup>1</sup>Sistema electrónico de entretenimiento para el hogar que ejecuta juegos electrónicos, contenidos en cartuchos, discos ópticos, tarjetas de memoria o cualquier dispositivo de almacenamiento.

como este proyecto trata de dar a conocer algunas pautas de cómo la educación y los videojuegos pueden ir de la mano y ser útiles.

# CAPÍTULO 1

## PLANTEAMIENTO DEL PROBLEMA

### 1.1 Antecedentes

La Historia es la ciencia que estudia el pasado de la humanidad y de los pueblos en sí, esto es realmente importante ya que así se puede comprender plenamente la naturaleza y el propósito de la existencia humana y muchas veces no se le da la importancia que se merece.

En la actualidad es una de las asignaturas estudiantiles que se encuentran entre las más complicadas para el proceso de enseñanza-aprendizaje, dado el gran volumen de información que abarca, además, en algunos casos se utilizan métodos que podrían considerarse obsoletos y poco eficientes (por ejemplo la memorización simple de fechas o nombres de personajes influyentes de una época), creando en el joven aprendiz un rechazo natural por una materia que puede llegar a considerarse aburrida y poco interesante, sin percatarse de lo esencial que significa para cualquier ser humano el conocer su propia historia y en especial para las nuevas generaciones que serán los responsables de guiar el destino de este país en el futuro. Esto se convierte en un problema gigantesco dado que la Historia no solo relata sucesos o acontecimientos pasados, sino que pone de manifiesto los posibles errores o aciertos que se dieron en una determinada época, si se aprende de estos acontecimientos se tendrá una herramienta muy poderosa para el desarrollo de la humanidad en el futuro. Se debe recordar que “el ser humano es el único ser que se tropieza con la misma piedra dos veces”<sup>2</sup>, el conocer el pasado y aprender de él da la posibilidad de no cometer un mismo error.

Pero éste no es un problema que afecta solo a los ecuatorianos, se reproduce en prácticamente todos los países del mundo. De acuerdo al desarrollo tecnológico que se ha dado en el cine, televisión y comunicaciones, este problema ha sido combatido utilizando, entre otras cosas, la tecnología que en muchos casos ha

---

<sup>2</sup> Dicho popular.

venido a ser la base primordial de estos proyectos; un ejemplo hasta ahora utilizado es el desarrollo de programas televisivos (series animadas, películas, documentales, etc.) que son presentados a la población y se convierten en una forma de hacer que un pueblo aprenda de su historia, éste método ha tenido muy buenos resultados sin embargo se han evidenciado algunos problemas.

La mayoría de producciones televisivas han sido desarrolladas por estudios foráneos y presentan una “historia” que no necesariamente tiene una relación con el público del Ecuador, por ejemplo; muchas veces se presentan series animadas que muestran hechos como las guerras estadounidenses en las que los soldados norteamericanos pelean por sus tierras y los ideales de libertad, que dichos ideales nacieron en América del Norte y que gracias a ellos, se extendió por toda América, creando en todos los jóvenes que ven este tipo de trabajos una falta de identidad y muchas veces rechazo a sus propias raíces. Lo que se pretende es no sobreexponer la historia propia de algún país o región, es más se respeta mucho estos procesos libertarios, pero en la realidad, y hablando más concretamente de América del Sur, esta revolución (histórica Norteamericana) no tiene ninguna vinculación con los pueblos y es intención del presente trabajo el poder crear un medio por el cual se puedan utilizar herramientas tecnológicas propias de este tiempo para presentar la historia de nuestra región y específicamente la del Ecuador.

En la actualidad una área tecnológica que ha tenido un desarrollo increíble es la del ocio digital (videojuegos, internet, aplicaciones móviles, etc.) especialmente en el área de los videojuegos (La industria de los videojuegos, 2011) y la propuesta de este escrito es utilizar herramientas de última generación para crear un videojuego educativo que cuente una historia propia del pueblo ecuatoriano.

## **1.2 Formulación del problema**

En el último tiempo se ha evidenciado una disminución en el interés por el estudio la Historia del Ecuador, debido a la cantidad de información y los métodos involucrados, que no siempre han demostrado ser adecuados; tal es el caso de la memorización excesiva.

Existen medios disponibles para el aprendizaje, pero en muchos casos son basados en información histórica de otras sociedades, esto genera confusión y en ocasiones un sesgo en la interpretación de las raíces culturales de un pueblo.

Estas iniciativas a pesar de tener buenas intenciones, no siempre tienen el efecto deseado, ya que al ser masivas, no garantizan un aprendizaje puntual y personal.

## **1.3 Justificación**

El desarrollo de nuevas tecnologías para la educación ha tomado en los últimos años una gran importancia, dados los grandes resultados que se han producido entre la población; el hecho de que muchas instituciones apliquen dichas herramientas a las clases impartidas en sus aulas es una prueba fehaciente de esto.

Un videojuego puede ser considerado una herramienta más para la educación, gracias a la Internet el número de usuarios que puede tener un proyecto como el presentado se incrementa de una manera exponencial, y puede ser una puerta de enlace entre niños y jóvenes que al haber nacido en países diferentes (gracias al fenómeno de la migración) no pueden tener un contacto directo con la cultura propia de sus padres o familiares y que por ende también les pertenece; el crear un videojuego que implemente de una manera didáctica y precisa partes importantes de la historia y cultura ecuatoriana y de esta forma fomentar el amor y respeto por el país.

Además, se pretende dejar una base para que en un futuro las empresas de este sector de la industria tecnológica, puedan considerar el desarrollo de estos

productos que cuentan con un alto crecimiento a nivel mundial- dentro del país y de la oportunidad de cambiar un concepto errado sobre el desarrollo de videojuegos en nuestra sociedad y el de poder dar a las nuevas generaciones, más posibilidades de encontrar una especialidad con la que puedan mostrar toda su capacidad de creación y de invención, dado que en esta industria es donde más notorias son estas habilidades.

Esto define en sí, cuán importante y provechoso es para la sociedad y para la propia Universidad Politécnica Salesiana, el estudio de dicho tema ya que se ha expuesto en pocas palabras el aporte social, educativo y económico que puede generar el desarrollo de este tema.

## **1.4 Objetivos**

### **1.4.1 Objetivo general.**

Desarrollar un prototipo de videojuego que apoye significativamente los procesos de enseñanza-aprendizaje de los hechos históricos más importantes de Quito y del Ecuador, a través de la integración didáctica y consistente de momentos relevantes del periodo comprendido entre 1500 y 1533.

### **1.4.2 Objetivos específicos.**

- Investigar acerca de todos los aspectos didácticos aplicables a los videojuegos para proveer al producto de una interacción atrayente para el jugador (alumno).
- Investigar y sistematizar hechos relevantes de la historia de Quito y del Ecuador que puedan ser incluidos en los relatos y ambientes del videojuego.
- Crear la historia y los guiones para cada uno de los niveles, los cuales deben ser consistentes y graduales entre sí, en base a los hechos investigados, considerando los aspectos de la didáctica para este tipo de aplicaciones.



- Implementar las secuencias, imágenes, ambientes y personajes del videojuego; todos asociados al guión.
- Diseñar e implementar los mecanismos requeridos para la interacción jugador-videojuego.
- Integrar y probar convenientemente el videojuego, de modo que signifique una herramienta didáctica y adecuada para aprendizaje.
- Incrementar el interés de las personas por el aprendizaje de la historia del país.
- Apoyar la difusión de la cultura e historia en personas que se encuentran fuera del territorio ecuatoriano.
- Contribuir al crecimiento de la industria del desarrollo de videojuegos dentro del Ecuador.

## **1.5 Metodología**

### **1.5.1 SCRUM.**

Desarrollada por Jeff Sutherland y Mike Beedle, es una metodología ágil para gerenciar y controlar el desarrollo de software en forma iterativa e incremental, que puede ser usada para manejar el desarrollo de productos complejos de software. (Letelier, 2006)

SCRUM se puede interpretar de la siguiente manera:

- “Un marco de trabajo o framework que sirve para agilizar el desarrollo de software y asegurar que lo más valioso del producto se termine en el tiempo más corto posible.”
- “Compuesto por equipos multifuncionales y auto administrados encabezados por un ScrumMáster que trabajan en ciclos conocidos como Sprints.”
- El equipo de desarrollo selecciona el trabajo a realizar de una lista conocida como Product Backlog que no es más que un listado de requerimientos priorizado por el dueño del producto en base al valor que cada requerimiento da al negocio.
- En cada sprint se lleva a cabo una reunión de planeación o también conocida como Sprint Planning Meeting, donde se analizan los requerimientos más prioritarios y se estima el esfuerzo necesario para completarlos.
- Del Product Backlog, el equipo selecciona los requerimientos prioritarios que puede desarrollar completamente en 30 días y define las tareas necesarias para cumplir con esos requerimientos; ese listado se lo conoce como el Sprint Backlog.

- Al final del sprint el equipo de desarrollo hace una demostración del software que termino al Dueño del producto y cualquier otro interesado del proyecto, esta demostración se conoce como "Sprint Review Meeting". (Estrella, 2007)

### **Beneficios**

- Se obtiene software lo más rápido posible y este cumple con los requerimientos más importantes.
- Se trabaja en iteraciones cortas, de alto enfoque y total transparencia.
- Se acepta que el cambio es una constante universal y se adapta el desarrollo para integrar los cambios que son importantes.
- Se incentiva la creatividad de los desarrolladores haciendo que el equipo sea auto administrado.
- Se mantiene la efectividad del equipo habilitando y protegiendo un entorno libre de interrupciones e interferencias.
- Permite producir software de una forma consistente, sostenida y competitiva. (Estrella, 2007)

### **1.5.2 Metodología SCRUM para creación de videojuegos.**

La metodología utilizada en este proyecto es SCRUM para videojuegos tiene como objetivo desarrollar videojuegos de calidad en tiempo y costo, así como la mejora continua del proceso para incrementar su eficacia y eficiencia. Pretende obtener resultados predecibles, administrar eficientemente los recursos y riesgos del proyecto, y lograr una alta productividad del equipo de desarrollo.

SCRUM fue concebido para que se adapte a equipos multidisciplinarios pequeños (de tres a siete integrantes que trabajan en un mismo lugar físico o

estén distribuidos), y para proyectos cortos (menores a un año de duración) con alto grado de participación del cliente.

#### ***1.5.2.1 Roles.***

La metodología define cuatro roles: equipo de desarrollo, productor interno, cliente y verificador Beta.

El equipo de desarrollo tiene las características que definen sub-roles dentro del equipo. Estos se corresponden con los que se utilizan habitualmente en la industria local de videojuegos y son los de programador, artista gráfico, artista sonoro y diseñador de juego.

El rol de verificador Beta no está presente en SCRUM pero si se detecta su existencia en el relevamiento de la realidad local y en la industria del videojuego en general. (Acerenza & Mesa, 2009)

#### ***1.5.2.2 Ciclo de Vida.***

Existen cinco fases secuenciales que son:

- Concepto
- Planificación
- Elaboración
- Beta
- Cierre

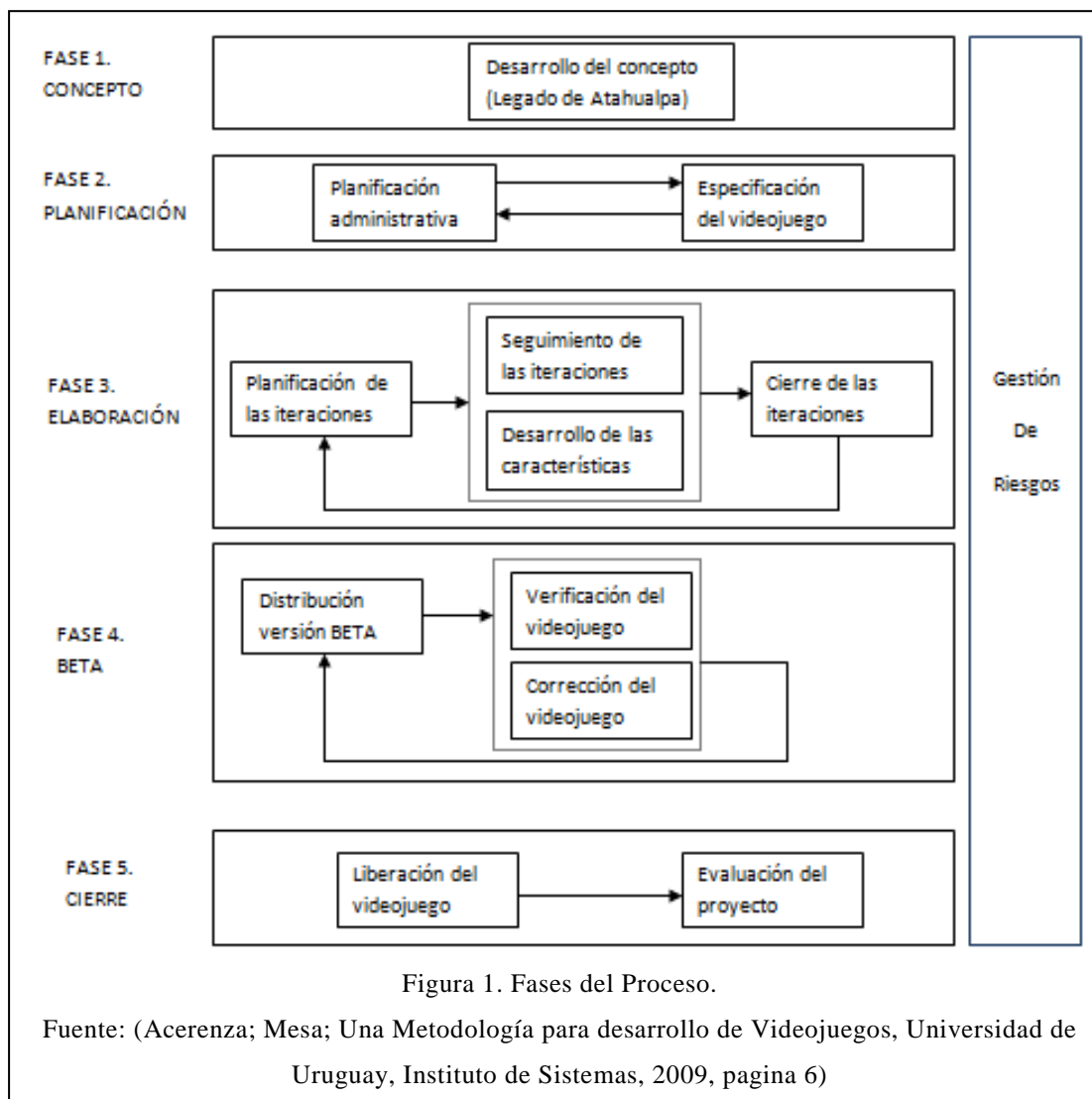
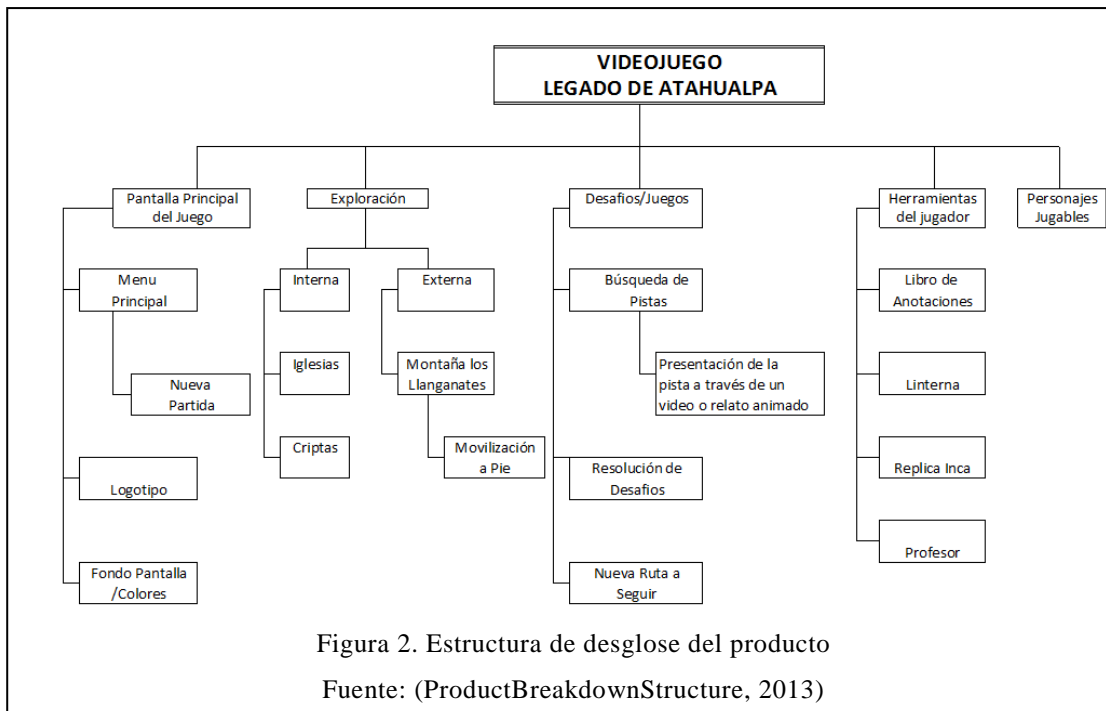


Figura 1. Fases del Proceso.

Fuente: (Acerenza; Mesa; Una Metodología para desarrollo de Videojuegos, Universidad de Uruguay, Instituto de Sistemas, 2009, pagina 6)

## 1.6 Alcance



El título del videojuego es “El Legado de Atahualpa”, el objetivo principal del jugador es encontrar el tesoro que escondió el General Inca Rumiñahui, éste sirvió como rescate del emperador Atahualpa que se encontraba secuestrado por los conquistadores españoles, pero al ser asesinado el Emperador Inca en manos de sus captores llevó al General Rumiñahui a tomar el tesoro y esconderlo en un lugar hasta ahora secreto.

Este proyecto se prevé que funcione en una PC con sistema operativo Windows, los requerimientos mínimos serán:

- Sistema Operativo Windows XP o posterior.
- 50 Gigabytes de espacio en Disco Duro.
- 1 Gigabyte de memoria RAM.

La instalación se la realizará a través de un archivo ejecutable (Legado\_Atahualpa.exe).

En el videojuego se controlará a un personaje principal el cual deberá recorrer iglesias del Centro Histórico de Quito, este personaje será un estudiante de arqueología con un gran conocimiento acerca de la historia de América del Sur y mediante pensamientos propios (deducciones de pistas, aclaración de puntos importantes de la historia del Ecuador) dará visos al jugador para resolver los diferentes desafíos, que estarán relacionados con la historia de Quito y del Ecuador, ya que sin estos conocimientos sería muy difícil el poder resolverlos.

Un ejemplo de interacción que se tendrá en el producto será la resolución de un rompecabezas, que cuenta la historia de la creación del mundo desde la perspectiva de los Incas, mediante imágenes dibujadas por la civilización Incaica y las cuales deben estar en orden para poder abrir una cripta dentro de una iglesia.

#### **1.6.1 Aspectos PBS.**

El videojuego constará de dos partes principales que son:

- Exploración.
- Desafíos/Juegos.

En el área de Diseño constará de tres partes importantes, que son:

- Pantalla Principal del Juego.
- Herramientas del Jugador.
- Personajes Jugables

#### ***1.6.1.1 Pantalla principal del juego.***

En este apartado se presenta tanto el logotipo como la pantalla de Inicio del videojuego con la opción:

Nueva Partida, que se refiere a comenzar el juego desde el inicio.

#### ***1.6.1.2 Exploración.***

Este tema se refiere a la forma en que el jugador se movilizará y actuará en los diferentes ambientes o escenarios que se encuentren en el videojuego.

La exploración interna se refiere a la movilidad que se tendrá dentro de edificaciones o lugares cerrados así como también la interacción que se poseerá con los diferentes objetos que existirán dentro de estos lugares, pero que al mismo tiempo se diferenciarán entre sí; es decir esta tiene que ver con que la exploración que se posea en una iglesia o edificio será diferente a la que se tendrá en una cripta o cueva.

La exploración externa implica el cómo se movilizará el jugador en las afueras de una montaña en busca del templo Inca

#### ***1.6.1.3 Desafíos / Juegos.***

Aquí se desarrollarán los diferentes desafíos que tendrán interacción, el jugador deberá superarlos para avanzar en el videojuego y estarán integradas al mismo.

La búsqueda de pistas será el juego principal a superar, es decir, el personaje principal deberá encontrar pistas sobre el tesoro de Atahualpa y para ello deberá encontrar rastros del mismo, los cuales estarán protegidos por trampas o rompecabezas dejados por el General Inca Rumiñahui; cada vez que se encuentre una pista se mostrarán imágenes vinculadas a los hechos históricos que proveerá una ruta a seguir.



Pero para conseguir dichas pistas se deberán resolver juegos más pequeños como por ejemplo abrir criptas o cuevas a través de la resolución de un rompecabezas de placas incaicas, a esto se refiere la resolución de desafíos/juegos.

#### ***1.6.1.4 Herramientas del jugador.***

Se crearán y modelarán las herramientas que tendrá el personaje principal del juego, con el cual el jugador podrá revisar las pistas recolectadas, resolver los diferentes mini-juegos y guardar datos importantes de la historia; como ejemplo de esto será un libro de anotaciones en donde se relatarán los diferentes acontecimientos que se implementarán y sucederán en el juego.

Las herramientas que podrá utilizar el jugador serán:

- Una linterna la cual se conseguirá con el pasar del juego
- Una llave Inca.
- Un libro de anotaciones en donde se guardará hechos importantes de la historia Quito y del Ecuador y que servirá como una herramienta de consulta y también servirá para tener acceso al personaje secundario el cual dará pistas para avanzar en el juego

#### ***1.6.1.5 Personajes jugables.***

Se modelará el personaje principal y un personaje secundario que intervendrán en el videojuego, así como su integración en el ambiente principal del juego que será la ciudad de Quito, específicamente el Centro Histórico.

### **1.6.2 Cambios realizados en la PBS.**

Se efectuaron cambios en la PBS, esto para que no afecte la estructura del videojuego, uno de ellos fue la exploración externa la cual se realiza en montañas, este cambio se lo explica de una mejor manera en el Capítulo 3 en la creación del guion, otro punto importante que se debió reconsiderar fue la opción

de guardar la partida, esto debido a que es necesario que el jugador mantenga el interés en el videojuego, ya que al ser corto (entre 20 y 30 minutos de duración) se puede obviar el guardado para que el jugador termine completamente el juego.

## **CAPÍTULO 2**

### **MARCO TEÓRICO**

#### **2.1 Concepto de Prototipo en Ingeniería**

Los prototipos son una representación limitada de un producto, permite probarlo en situaciones reales o explorar el uso, creando así un proceso de diseño de iteración que genera calidad. Un prototipo puede ser cualquier cosa, desde un trozo de papel con sencillos dibujos a un complejo software. (Lacalle, 2006)

En el desarrollo de videojuegos existe el prototipo Alfa, que no es más que la primera versión del programa, esto se refiere a una fase del producto que todavía es inestable y que necesita ser pulida para eliminar errores, pero esta satisface la mayoría de requisitos. Existe también la versión Beta la cual es la primera versión completa de un programa o producto, es posible que sea inestable pero útil para demostraciones previas a usuarios. (Pincay Ponce, 2008, 5-8)

En ingeniería este término se refiere a los primeros equipos que se realizaron en laboratorios de desarrollo, puede ser efectuado en plan artesanal los cuales pueden contener varias modificaciones que los diseñadores incluyen según avanzan con el tiempo.

#### **2.2 Concepto de videojuego**

Un videojuego se lo definiría como un programa creado para divertir y/o entretener, basado en la interacción entre una o varias personas y un aparato electrónico; los cuales recrean entornos virtuales para que el jugador controle a un personaje ficticio o cualquier otro elemento que se encuentre relacionado con el entorno en el que se desarrolla con un objetivo y reglas determinadas. (Gil & Vida, 2007)

## **2.3 Clasificación de los videojuegos**

De acuerdo a la estructura y según las principales habilidades que utiliza el jugador, existe la siguiente clasificación (Hernández, 2010):

Arcade: “Juego de acción caracterizado por la rapidez de respuesta requeridas al usuario, con una sucesión de metas o etapas previas para alcanzar el objetivo final”. Ejemplos: Donkey Kong, Sonic, Super Mario Bros.

Shot'm up: “Uso explícito de armas y violencia, normalmente en un entorno tridimensional.” Ejemplos: Doom, DukeNukem, Unreal.

Simuladores: “Juegos que intentan emular situaciones, acciones o funcionamientos derivados de la vida real. Deportivos (FIFA), Juegos de mesa y recreativos; el ordenador puede sustituir a otro jugador e incorporar nuevas funciones (Monopoly), vehículos (Apache), lucha (Street Fighter), sociales (SimCity).”

Rol: “El jugador encarna a un personaje asumiendo y desarrollando su personalidad e interactuando con el resto de los elementos; se caracteriza por el control exhaustivo de las constantes vitales, habilidades y otras características propias del personaje.”

Aventura gráfica: “Predomina los diálogos y la interacción con los otros personajes y elementos. Suelen ser lineales e incluir la superación de enigmas, problemas lógicos, Ejemplos: Monkey Island, Indiana Jones.” (Clasificación de videojuegos, 2010, 1-2)

## **2.4 Desarrollo y elaboración de videojuegos**

Es una actividad en la cual se diseña y produce un videojuego, la cual involucra profesionales de la informática, diseño, sonido, actuación, programación, desarrollo, etc.

El proceso que generalmente se utiliza es:

- Concepción de la idea del juego: la cual plantea la estructura de la narración, el proyecto realizado se basa en narraciones existentes donde se plantea en qué consiste el juego y que se pretende que realice el jugador.
- Diseño del storyboard: se plantea la estructura del juego, se visualiza gráficamente para representar la evolución de la narración y del jugador dentro del juego.
- Programación del comportamiento del videojuego: esto corresponde a la estructura del programa, se plantea los requerimientos del sistema.
- Desarrollo del arte: se refiere al diseño y desarrollo de ambientes y personajes involucrados dentro de la historia que requiere el juego; teniendo en cuenta el tamaño del software de acuerdo a los requerimientos del sistema.
- Producción de sonido: se enfoca en la producción de efectos de sonido para complementar la parte visual del videojuego, esto con el fin de generar emociones en el jugador para permitir una mejor interacción de este con el juego.
- Testing: una vez terminado el juego, se requiere comprobar que todo esté en orden y funcional, esta fase permite corregir detalles o problemas que se necesiten
- Mecanismo del juego: Consiste en crear, verificar y realizar las reglas básicas e interacciones que hacen que un juego sea entretenido, entender esto es lo más importante para un buen diseñador de videojuegos, se debe tener en cuenta que al realizar una buena historia con sus respectivos personajes, combinados con mecanismos sólidos es importante en la

generación del juego, con el fin de lograr un videojuego entretenido. (Lacassa, 2011)

Un buen juego combina los mejores mecanismos así como gráficos, personajes y música que sean llamativos; sin embargo estos aspectos no son lo único para hacerlos entretenidos ya que los buenos videojuegos pueden jugarse sin tener todas estos aspectos. (Lacassa, 2011)

Se debe también tener en cuenta dentro del mecanismo los retos interactivos y la dificultad que estos tengan dentro del videojuego, ya que al ser interactivos, los jugadores tienen un tipo de control que les ayudara a estar expectantes; se debe tener claro que un reto debe ser lo suficientemente fácil para entender pero a la vez lo suficientemente difícil para superarlo, esto con el fin de que el jugador no se rinda fácilmente y continúe jugando el videojuego. (Lacassa, 2011)

## **2.5 El guión**

Es un texto en que se presenta, con todos los detalles necesarios para su elaboración, puede ser para una película, un programa de radio o televisión, o un videojuego. El cineasta Julián Lara sostiene que el guión es una pauta en la que aparece descrito todo lo que después saldrá en la pantalla.

Existen varios tipos de guiones a considerar:

- Guión cinematográfico

“Es aquel que describe lo que se mostrará y escuchará en una obra cinematográfica.” Involucra división por escenas, acciones de los personajes o eventos, diálogo entre personajes, así como breves descripciones del entorno. Un buen guión literario tiene que transmitir la información suficiente para que el lector visualice la película, cómo transcurre el diálogo, cómo actúan los personajes y con qué objetos interactúan. (La noche interminable. Investigación, 2010)

- Guión literario

Es un texto que contiene una historia que ha sido pensada para ser narrada en imágenes y sonido, se escribe de manera clara sin dar indicaciones técnicas para la realización. (Radulescu, 2012)

- Guión técnico

Es un documento de producción que contiene la información necesaria para desarrollar cada uno de los procesos que la obra audiovisual requiere.

- Guión videojuegos

Es muy similar a los realizados para libros y películas. Este guión puede estar condicionado por las acciones del jugador resultando en múltiples situaciones basadas en dichas acciones, complicando exponencialmente su desarrollo.

Hay que tomar en cuenta el argumento de la historia, es decir la idea que se tiene para desarrollar de una manera versátil, el desenvolvimiento de lo que se va a relatar o contar. Lo más conveniente es tener un único guionista, que redacta el guión, y luego pasarlo al dibujante para que realice bocetos de lo que va a ser el guión, claro esto si es que no son la misma persona.

## **2.6 Investigación y análisis**

Lo principal en la investigación fue seleccionar una leyenda que pueda ser convertida en videojuego y que sea de interés; en este caso fue “La leyenda del tesoro de Atahualpa”.

Los videojuegos de hoy en día normalmente para terminarlos llevan de uno a cuatro años de desarrollo con equipos de entre 10 a 40 personas con experiencias en diferentes campos como diseñadores, programadores, técnicos de audio,

dibujantes, etc. junto con presupuestos millonarios para generar videojuegos de buena calidad y divertidos, utilizando tecnología bidimensional y tridimensional.

Por este motivo al momento de escoger la plataforma a utilizarse se tuvieron en cuenta las siguientes características:

- Que la plataforma no requiera soporte sofisticado, esto para que cualquier persona pueda acceder a este como a los ejecutables.
- Que la plataforma sea amigable tanto en interfaz como en herramientas para trabajar de una manera sencilla.
- Considerar el poder comercializar los productos finales y para esto el software no debía tener limitantes en requerimientos de autor.
- El tiempo de aprendizaje del software así como de sus herramientas sea mínimo en consideración a otros programas para desarrollo de videojuegos.

### **2.6.1 Unity.**

Es un motor de videojuegos multiplataforma de desarrollo y permite la integración con programas tales como 3D Max, Maya, Softimage, Blender, Adobe Photoshop; los cuales son utilizados para diseñar y mapear estructuras que después se incorporaran al videojuego. Como motor gráfico utiliza Direct3D para Windows, Open GL (Mac, Linux). Tiene soporte de mapeado, sombras, renderizado y textura esto con el fin de realizar efectos de procesamiento en pantalla completa.

El scripting se basa en Mono, el cual es una implementación de código abierto de .NET Framework, esto permite utilizar UnityScript (lenguaje personalizado inspirado en la sintaxis ECMAScript), también está basado en C#, incluye



UnityAsset Server la cual es una solución de control de versiones para los assets y scripts, utilizando PostgreSQL como backend. (TechnologiesUnity, 2012)

### **2.6.2 Maya.**

Es un software de animación en 3D que proporciona un conjunto completo de funciones creativas para realizar en 3D animación por ordenador, modelado, simulación, renderización y composición dentro de una plataforma de producción sumamente ampliable. Ahora Maya incluye tecnología de visualización de última generación, flujos de trabajo de modelado más rápidos y nuevas herramientas para gestionar datos complejos. (Autodesk, 2012)

#### **Características**

- Sombreado y visualización de última generación en la ventana gráfica
- Flujo de trabajo de modelado poligonal más rápido
- Datos más avanzados con las herramientas de ensamblaje de escenas facilitando la creación de modelos grandes y complejos

## **CAPÍTULO 3**

### **DISEÑO DEL VIDEOJUEGO**

En todo proyecto de ingeniería es fundamental considerar para la construcción de un videojuego dos aspectos importantes, el primero tiene que ver con la capacidad psicomotriz de los individuos que se define como el desarrollo de los aspectos motores, manuales y de reflejos, el segundo punto a considerar es el diseño y estructura del juego que desarrolle interés y curiosidad frente a un mundo imaginario en el cual pueden interactuar e identificarse para socializarlo posteriormente.

#### **3.1 Creación del guión**

Dentro del proceso de creación del guión para el videojuego se determinaron los siguientes elementos, los cuales son tomados del autor Daniel González:

- Idea: El juego debía ser de tipo aventura rol. Su nombre “El legado de Atahualpa”.
- Descripción: “El legado de Atahualpa” es un juego en el cual el personaje debe atravesar 4 niveles; el primero se realiza en una iglesia, el segundo en una cámara secreta debajo de la iglesia, el tercero en una montaña, y el cuarto dentro de un templo inca. En cada nivel debe superar retos para pasar al siguiente nivel esto con el fin de encontrar un tesoro escondido. (González, 1995)
- Diseño: “El legado de Atahualpa”.
  - Objetos: en la iglesia habrá columnas activadas por un mecanismo secreto, en las cuales existen signos del sol y la luna que al ser presionados en el orden correcto abrirán el pasaje hacia la cámara secreta. Dentro de la cámara secreta existirán botones con imágenes que dan a conocer la leyenda del primer inca y al ser

presionados correctamente harán caer el sol inca el cual debe ser recogido. En la montaña existe un templo, el cual se debe buscar para ingresar. Dentro del templo existe una réplica del sol inca al cual se debe presentar el sol inca recogido, estos delimitan lo que el jugador debe realizar dentro del juego.

- Sonidos: los sonidos utilizados dentro del juego son varios tanto para la ambientación de los niveles, así como para el momento de caminar del usuario o en el caso de la montaña el sonido de agua y viento.
- Controles: Con el ratón se puede delimitar hacia donde mira el usuario utilizando un puntero circular que aparece en la pantalla, con el botón derecho aparecerán ayudas en diferentes partes de los niveles, y para moverse dentro de los niveles se utilizaran las flechas del teclado. (González, 1995)

### **3.1.1 Datos importantes sobre el guión.**

La diferencia entre una novela y un guión de videojuego es que la novela es una estructura narrativa cerrada, un videojuego es controlado por usuario a través de mecánicas de juego y hay que incitar a resolver ciertos enigmas y a crear situaciones jugables y divertidas. (Gonzalez, 2009)

Dependiendo del tipo de juego el guion es la parte principal. Por debajo de conceptos como gráficos, sonido, jugabilidad, etc. el núcleo es el guión. Desde definir el personaje principal, su historia, el mundo en el que se va a desenvolver, su evolución, los enemigos, las conversaciones, los escenarios, las secuencias cinemáticas. (Gonzalez, 2009)

En los juegos de primera persona, el guión, es el elemento de la narrativa que es tan maltratado por la industria de los videojuegos, que ha sido puesto a menudo al servicio de otros elementos que no son necesariamente tan importantes como

pueden ser el apartado gráfico o el sonido, está claro que, obviando el género deportivo, en juegos en primera persona o en un título de aventuras, la importancia del argumento es relativa y no es imprescindible para crear un buen videojuego.

En los juegos de roles, hay otros géneros donde la importancia del guión se manifiesta y admite pocos matices. Y es que el pasar horas y horas al frente de un grupo de personajes poco carismáticos o cuya historia o progreso no interese en lo más mínimo acaba resultando un punto en contra para la jugabilidad.

El RPG (juego en tiempo real) está plagado de grandes historias, y ha sido uno de los estilos de juego que más ha evolucionado en este sentido.

Siendo uno de los factores más importantes, en valoración de un videojuego, el guión, el desarrollo de los acontecimientos, la evolución de la historia, de los personajes, y a la vez transmitir emociones al jugador mediante la narrativa, hace que los cambios que se presentan con el pasar del desarrollo del videojuego sea muy importante para que los videojuegos tanto clásicos como contemporáneos, sigan en el corazón de los jugadores.

Se puede mencionar el caso de ‘Assassin’s Creed’, un juego con una documentación sobresaliente, una recreación histórica más que notable y una historia que podía gustar, pero el guion estaba bien contado. En este juego se olvidó el factor fundamental: la jugabilidad. Y de ese modo el jugador se encuentra con misiones y submisiones clónicas, que son un calco unas de otras y parece que no tienen más objetivo que hacer pasar el tiempo hasta el siguiente punto de inflexión en la trama. (Villalobos & Candil, 2009).

En el juego ‘Gears of War 2’, reconocido por crítica y público como uno de los mejores juegos en primera persona, se presenta una guerra en un futuro de ciencia ficción, entre los humanos y una raza salida del subsuelo, sin complicarse en los motivos; el guión, dentro de sus limitaciones, funcionaba. Al terminar el juego se reflexiona sobre lo que ha vivido el personaje, y se observa

que la historia está llena de hilos sueltos, con multitud de cosas que no quedan claras, pero parecen más debidas a la precipitación y el desconocimiento que al mal hacer. (Villalobos & Candil, 2009)

A la hora de narrar, es importante que el guión establezca alguna relación con lo narrado, en algunos casos se ignora completamente al jugador, en series de toda clases hacen esto, sin embargo, empobrece al guión, donde presuntamente se tiene un protagonista con unos ideales prefijados, esto no funciona porque no desean escoger ciertos elementos clave para continuar con lo narrado, eso puede ser confuso y desorientar, cuando esto sucede no tiene importancia, sin embargo si estas contando una historia y deseas que esto no sea relevante, entonces ¿Qué importancia tiene esto?.

En el caso de este trabajo se observó con el pasar del desarrollo del videojuego, que al incluir una exploración externa extensa como el Centro Histórico de Quito, el jugador se desenfocaría de la historia que se quiere dar a conocer, por tal motivo se implementó una exploración más llamativa que se da en una montaña, en este caso Los Llanganates, esto para que no se pierda el interés en la trama del videojuego y lo más importante, finalice el juego.

### **3.2 Definición de secuencias**

Siguiendo el proceso se planteó los niveles del juego y se definieron los prototipos de los niveles de la siguiente manera:

- Primer nivel: Empieza dentro de una iglesia en reconstrucción, la cual debe recorrer para verificar que debe realizar, encontrar un mecanismo para que algunas columnas se levanten del piso; en estas encontrara varios signos y al momento de acercarse y presionar con un orden específico sobre ellas estas descenderán y abrirán un pasaje hacia una cámara.

- Segundo nivel: Dentro de la cámara se debe recorrer para encontrarse con una pared con un gran sol inca y algunas imágenes sobre unos botones que cuentan la historia del primer inca, estos deben ser presionados siguiendo una secuencia para que el sol caiga y se lo pueda recoger.
- Tercer nivel: Este se desarrolla en una montaña, el objetivo aquí es encontrar el templo inca y entrar en él.
- Cuarto nivel: Dentro del templo se deberá buscar un cuarto que contenga el sol inca y a este presentar la llave en forma de sol que se recogió en el segundo nivel.

### **3.3 Creación de personajes y ambientes**

El o los personajes en un videojuego son parte principal en el mismo, ya que estos atraerán la atención del jugador, en este caso el juego está pensado con un jugador en primera persona es decir se tendrá una vista de pantalla completa observando el nivel y todas sus características sin necesidad de presentar alguna parte del personaje.

Durante el proceso de diseño, se realizaron algunos bocetos de posibles personajes jugables pero estos presentaban algunas complicaciones al momento de implementarlos como movimientos erráticos, para lo cual se consideró realizar un personaje en primera persona es decir que el jugador tome el rol del personaje jugable.

El tener personajes que no representen un esfuerzo mayor en modelado y desarrollo y que no signifiquen una carga mayor en recursos al equipo que se está probando hace que el juego se ejecute de una buena manera, por este motivo se decidió encontrar modelos 3D de personajes que no sean tan detallados y que ayuden al proyecto en el desarrollo y eficacia.

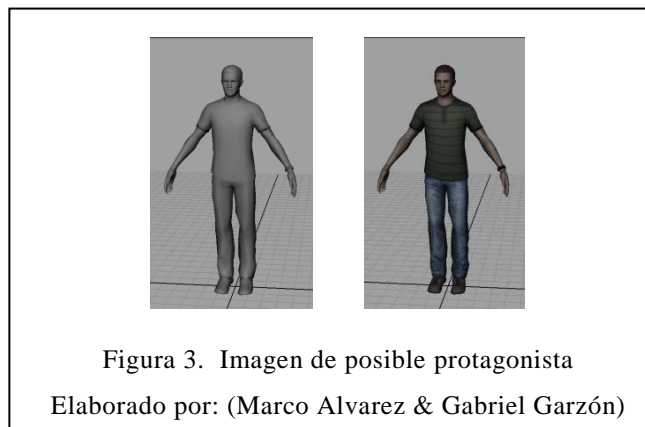
Gracias al desarrollo de comunidades de diseñadores gráficos, integradas en la mayoría por estudiantes, maestros y demás personas, se puede encontrar algunos de sus trabajos de versión libre y con licencia para utilizarlos solo como referencia para estudio, ya que a nivel de videojuegos es posible realizar esto, ya que es una práctica común para desarrollar videojuegos como en el caso de Grand Theft Auto en donde reutilizaron varios personajes de sagas anteriores para complementar el juego.

De este tipo de modelos se proveyó al proyecto, encontrando dos tipos de personajes.

- El personaje principal o protagonista.
- El personaje secundario.

### **3.3.1 Personaje principal o protagonista.**

Se establece como personaje principal a un joven no mayor de 22 años. Su estructura y modelo no suponen una carga mayor al esperado y compensa mucho en rendimiento. El modelo 3D del personaje fue exportado al modelador Maya<sup>3</sup> como un documento objeto (obj). Dicho formato permite manipular el modelo 3D en cualquier tipo de aplicación como Maya. Las siguientes imágenes Figura 3.1 pertenecen al modelo del personaje tanto en su estado raw como texturizado.



---

<sup>3</sup>Software de animación en 3D que proporciona un conjunto completo de funciones creativas para realizar en 3D animación por ordenador

### 3.3.2 Personaje secundario.

El personaje secundario se refiere a un profesor, el cual sirve de ayuda al personaje principal. El modelado de este personaje es mucho más sencillo ya que es un personaje que guía y que orienta al personaje principal dentro del juego.

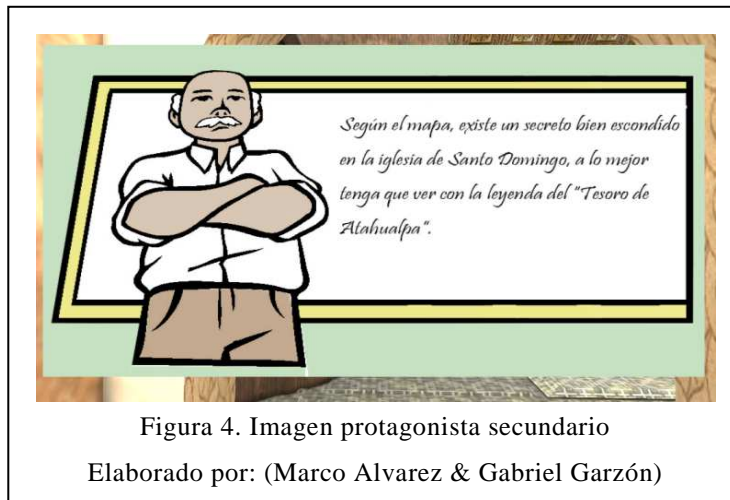


Figura 4. Imagen protagonista secundario  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

Otro punto a considerar es la ambientación del juego, para lo cual se revisó algunas historias sobre el tesoro de Atahualpa para verificar lugares donde se desarrolla y se decidió ambientarla en una iglesia, una cámara y en una montaña.

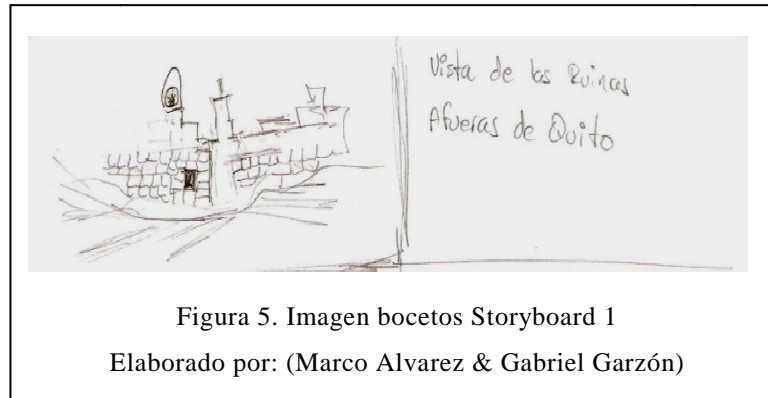
### 3.4 Creación del Storyboard

Realizado un boceto de personajes y ambientes, se inicia el proceso de desarrollo del Storyboard; el cual no es más que una serie de ilustraciones realizadas a mano o en un programa de dibujo con el objetivo de pre visualizar una animación.

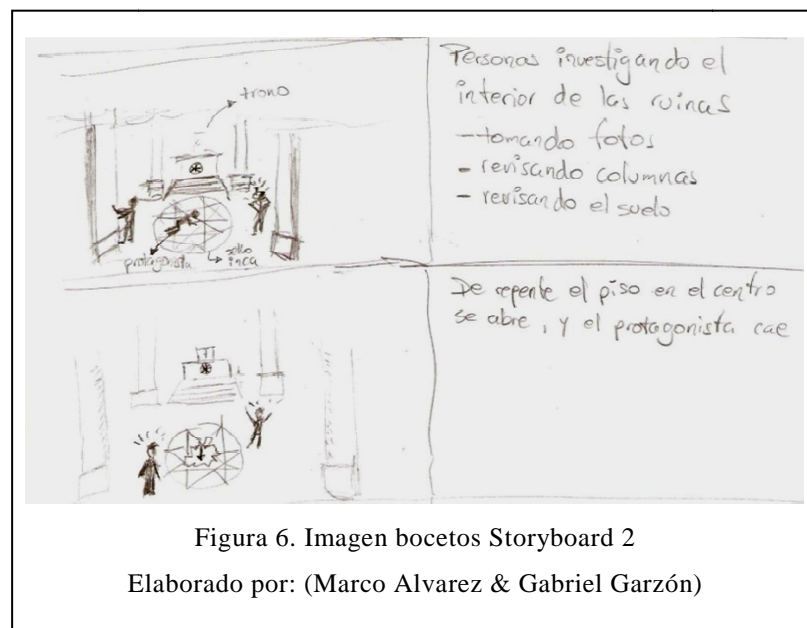
Para este proyecto se realizó un Storyboard con la descripción general de las escenas a visualizar en el videojuego. Las escenas preconcebidas a utilizarse se describen a continuación:



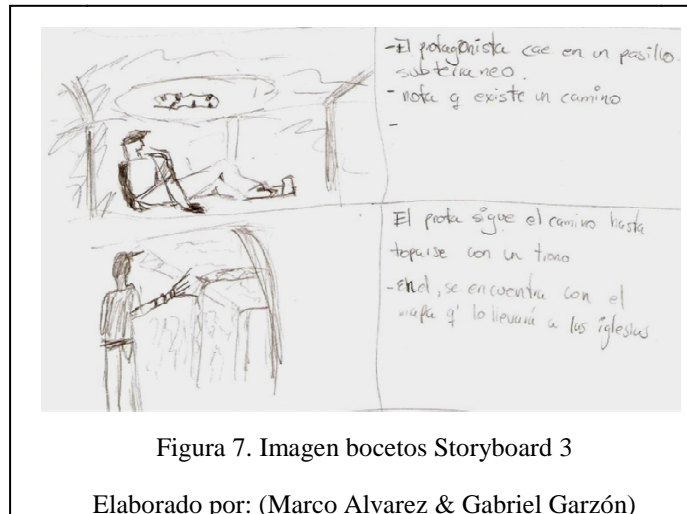
- Escena 1: Vista de Quito y algunas ruinas, la cual se sustituyó con un intro que explica como Atahualpa fue capturado en Cajamarca y como este ofreció su tesoro para que lo liberen.



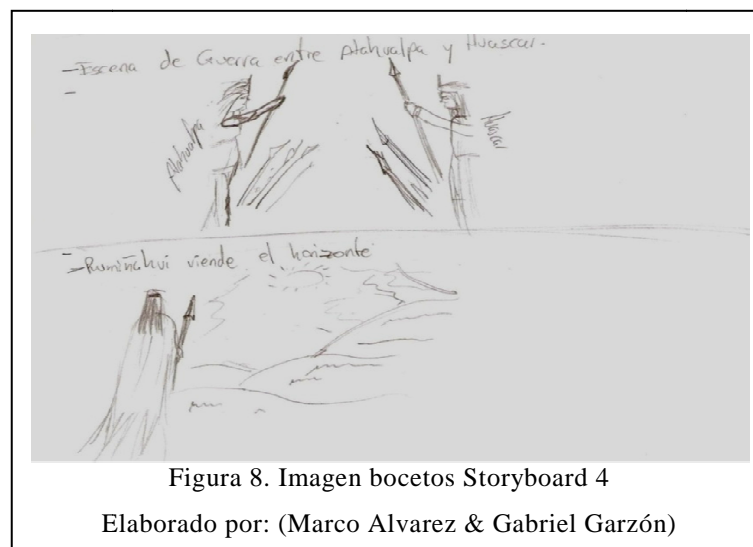
- Escena 2: Personas investigando en el interior de ruinas, este se cambió por una iglesia en la cual se debe investigar para encontrar una cámara secreta



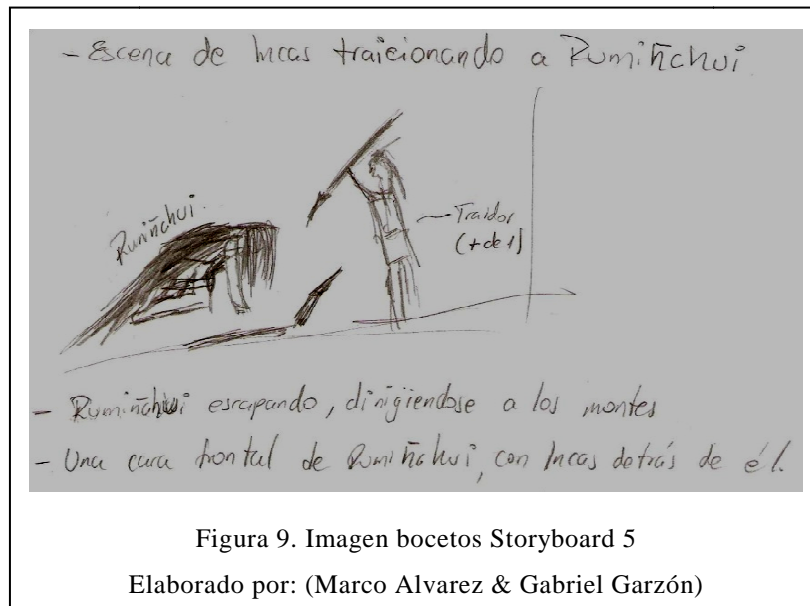
- Escena 3: Al caer en la cámara encuentra un camino y lo sigue para encontrar parte de un tesoro, se realizó el cambio de encontrarse con botones de piedra con imágenes para descubrir un sol inca.



Escena 4: Las escenas previstas eran de un encuentro entre Atahualpa y Huáscar explicando porque se dio la guerra entre pueblos y a Rumiñahui observando hacia el horizonte mirando la llegada de los españoles. En esta parte se decidió implementar niveles en los cuales es posible investigar y realizar ciertas acciones como la de explorar montañas y un templo inca.



- Escena 5: Se muestra la muerte de Atahualpa y como el oro es escondido, se realizó un gran cambio al implementar una animación final explicando que el tesoro no es en realidad oro, es la unión que exista entre pueblos para mantenerse juntos.



## CAPÍTULO 4

### DESARROLLO DEL VIDEOJUEGO

#### 4.1 Geometría 3D

El software escogido para el modelado de las estructuras (iglesia, cámara secreta, montaña, templo inca e interior del templo) fue, Autodesk Maya el cual es un software de creación de gráficos tridimensionales, animaciones y efectos especiales, además, de poseer varias herramientas para realizar modelado, render<sup>4</sup>, animación, simulación, etc.

Para el modelado de la iglesia, cámara secreta, templo e interior del templo se realizó varios cubos, utilizando las herramientas de modelado de Maya, se les fue dando dimensiones, además, se utilizó curvas y polígonos para recrear las estructuras.

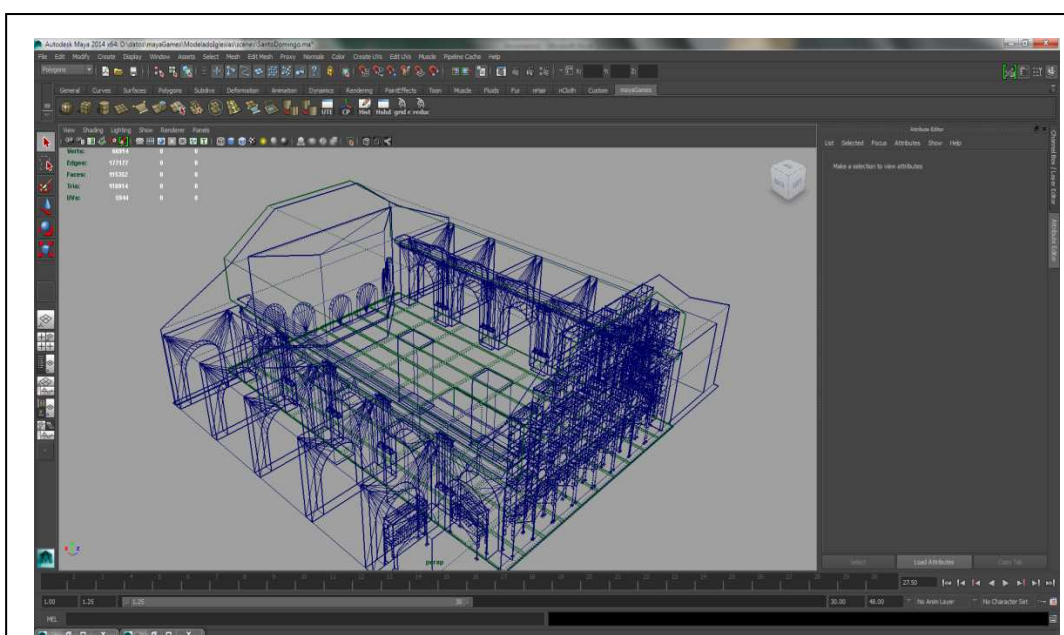


Figura 10. Geometría 3D de la iglesia

Elaborado por: (Marco Alvarez & Gabriel Garzón)

<sup>4</sup>Imagen digital que se crea a partir de un modelo o escenario en 3D realizado en algún programa de computadora especializado, cuyo objetivo es dar una apariencia realista desde cualquier perspectiva del modelo.

Maya permite observar en diferentes pantallas y ángulos el área que se está diagramando.

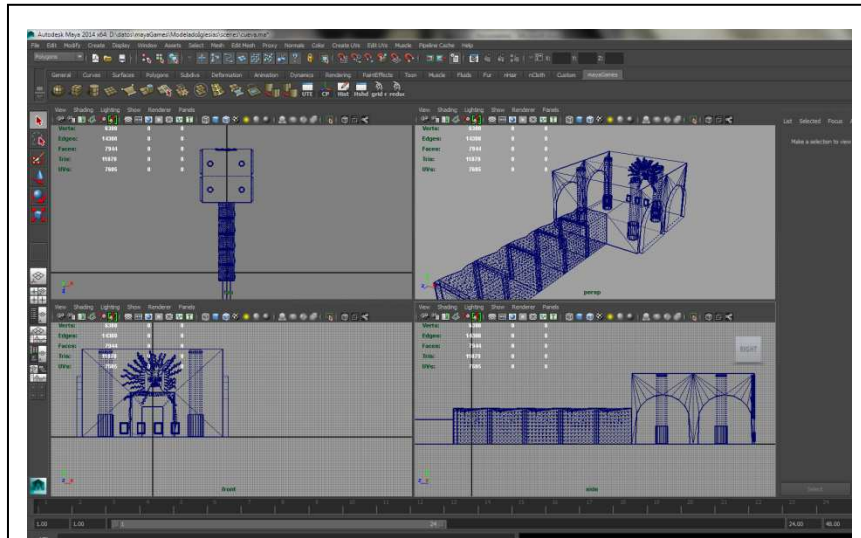


Figura 11. Geometría 3D de la cámara secreta, vista lateral y de frente  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

En la figura 12 se puede observar el diagrama de la geometría 3D del templo que se realizó, el cual muestra los niveles y alturas que esta tendrá.

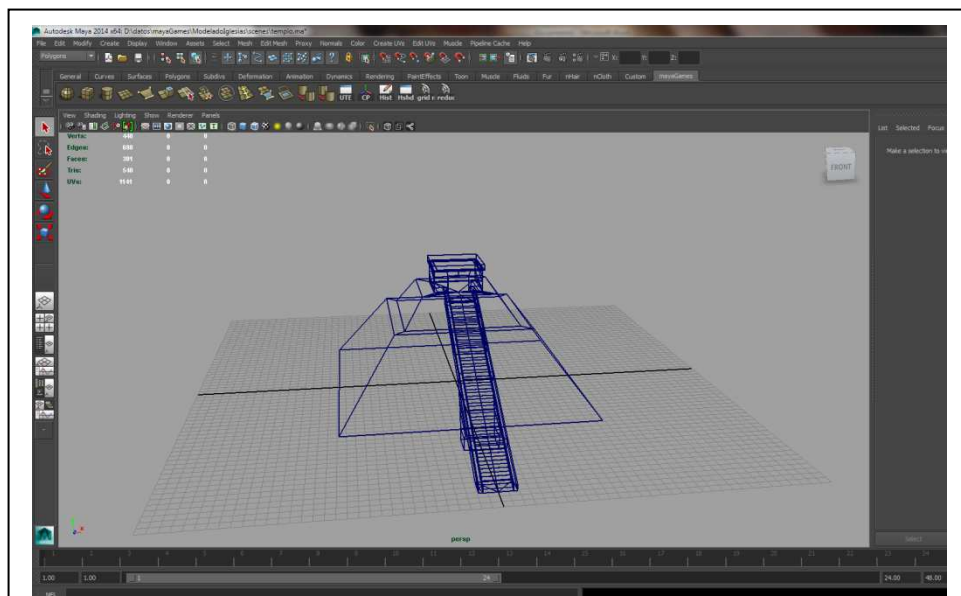


Figura 12. Geometría 3D del templo  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

## 4.2 Modelado

El modelado sirve para dar formas concretas a los modelos es decir crear el cuerpo que se generó con anterioridad en la geometría 3D, para esto en Maya se selecciona los polígonos y curvas realizados para asignarles con las herramientas de Maya un fondo gris, esto con el fin de realizar un refinamiento del modelo 3D antes realizado y así evitar que aristas, vértices y caras tengan aberturas al momento de pasar al texturizado.

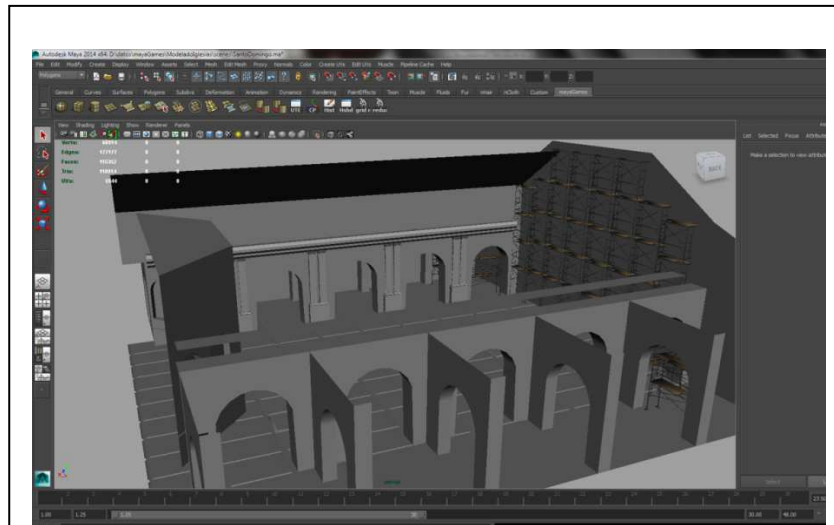


Figura 13. Modelado de la iglesia  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

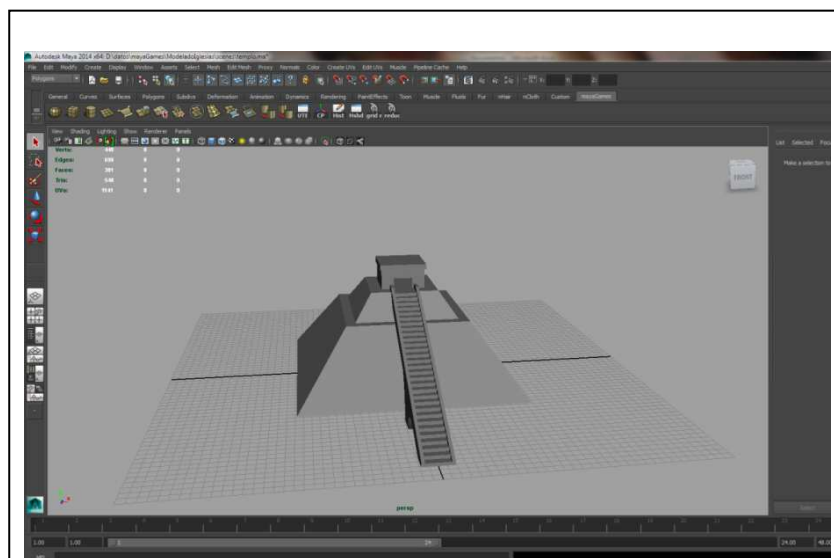


Figura 14. Modelado del templo  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

Vista del modelado de la cámara secreta ya con las formas definidas del pasaje y la cámara, junto con el sol Inca de la pared, se puede observar que las columnas tienen una mejor definición y además, se observa que los escenarios tienen un mejor acabado.

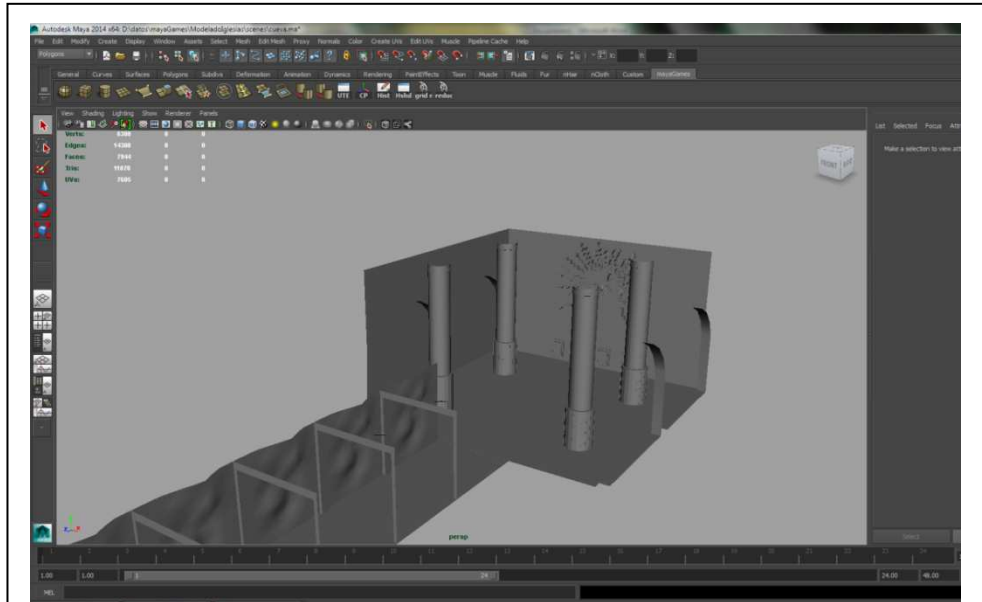


Figura 15. Modelado de cámara secreta  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

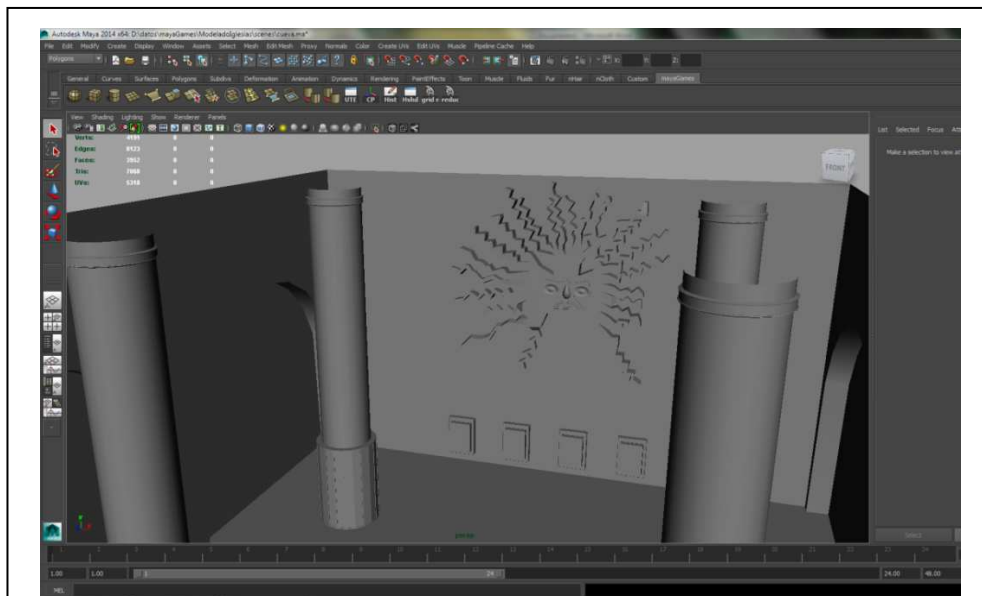


Figura 16. Modelado interno cámara secreta  
Elaborado por: (Marco Alvarez & Gabriel Garzón)



### 4.3 Texturizado de personajes y ambientes

El texturizado no es más que asignar los diferentes colores, aspectos y texturas al modelo generado con anterioridad, para esto se utilizó las herramientas de texturas que posee Maya, algunas de estas fueron exportadas con el fin de tener una mayor cantidad de opciones al momento de dar color e iluminación.

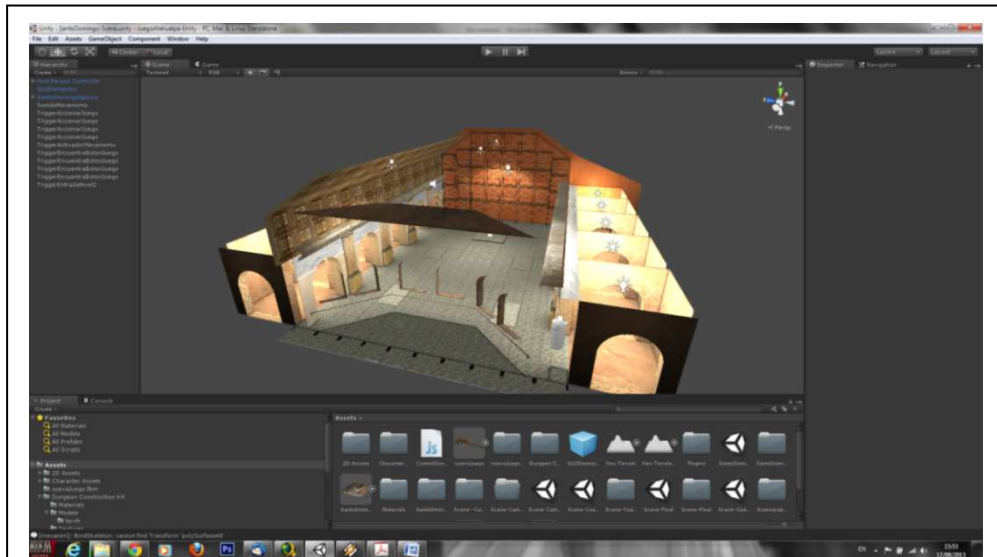


Figura 17. Texturizado Iglesia

Elaborado por: (Marco Alvarez & Gabriel Garzón)

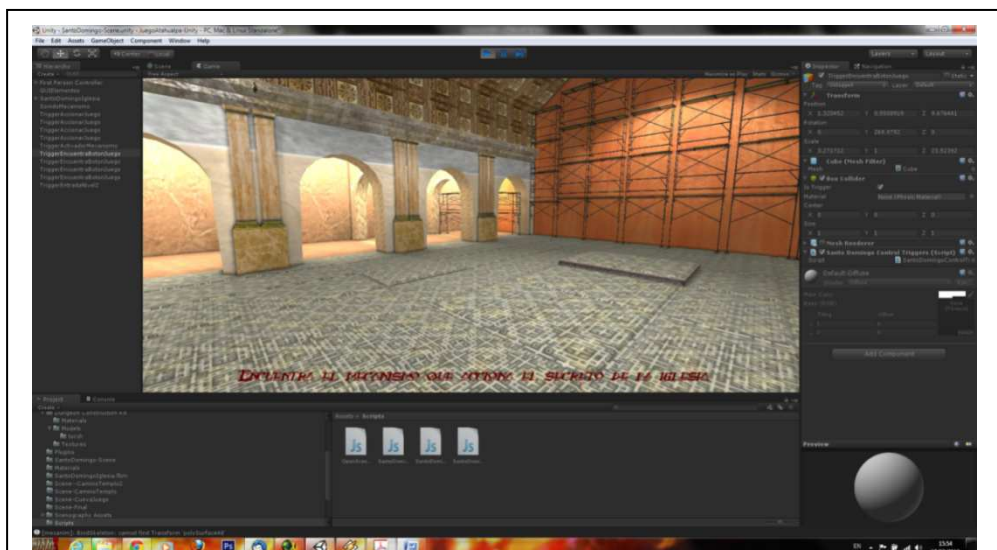


Figura 18. Texturizado interno Iglesia

Elaborado por: (Marco Alvarez & Gabriel Garzón)



Se debe revisar con detenimiento y desde varios ángulos como va quedando el texturizado de los diferentes niveles, esto con el fin de revisar que no quede ningún punto sin asignar color para que al momento de ejecutar la animación de la escena no de problemas.

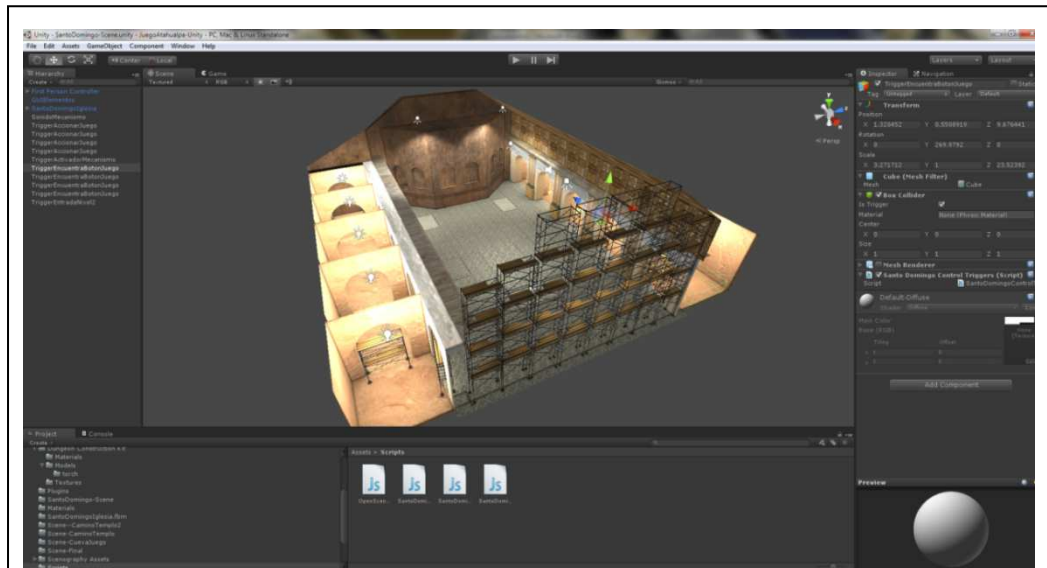


Figura 19. Texturizado interno y vista desde arriba Iglesia  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

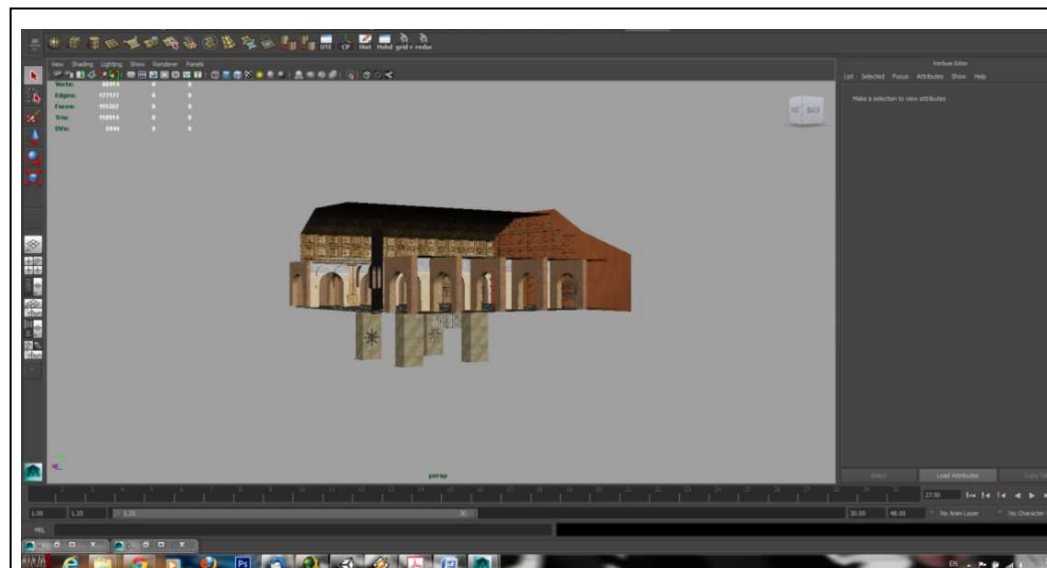


Figura 20. Texturizado interno y columnas que se levantan Iglesia  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

Se debe tener en cuenta no texturizar con colores oscuros y dar diferentes tonalidades en las imágenes para tener una mejor idea de cómo va a observar los niveles el jugador y que las diferentes estructuras y objetos no se pierdan al formar uno solo, es decir que cada uno tenga su estructura delimitada y estilizada.

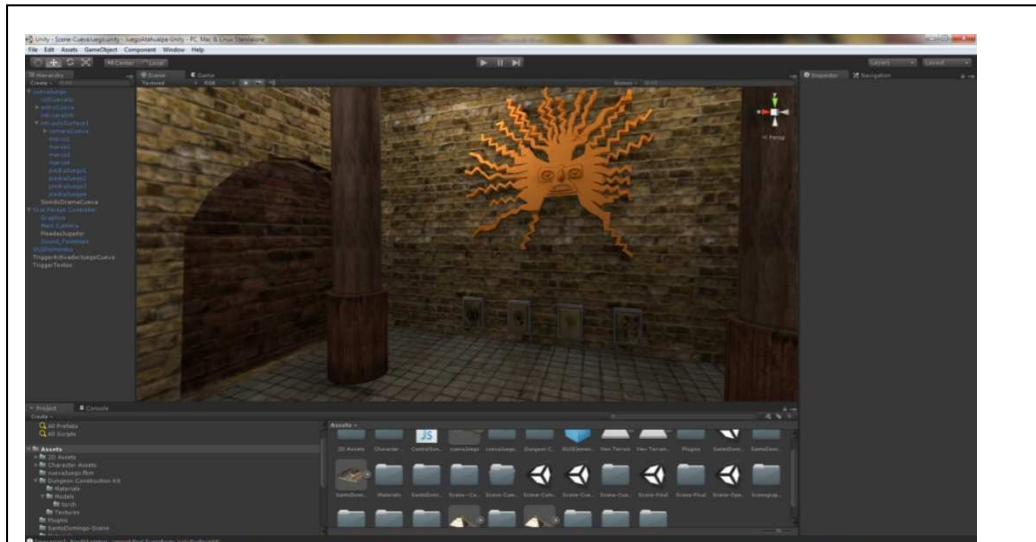


Figura 21. Texturizado interno cámara secreta, sol Inca  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

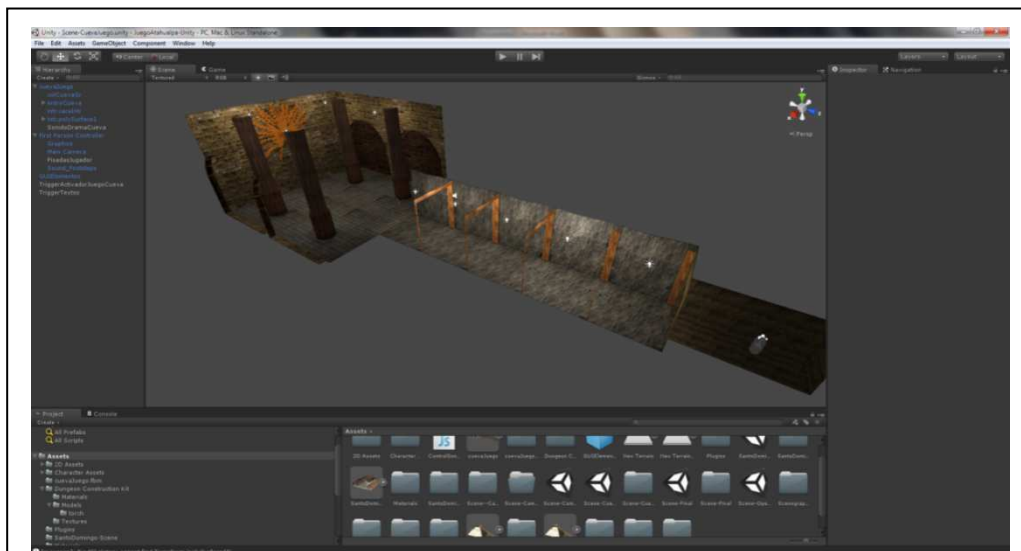


Figura 22. Texturizado interno cámara secreta  
(Elaborado por: Marco Alvarez & Gabriel Garzón)

Al igual que en las figuras anteriores es necesario dependiendo del nivel observar desde varios ángulos como va quedando las texturas utilizadas con el fin de encontrar fallos en los niveles y observar como la iluminación que se implementa sea la adecuada.

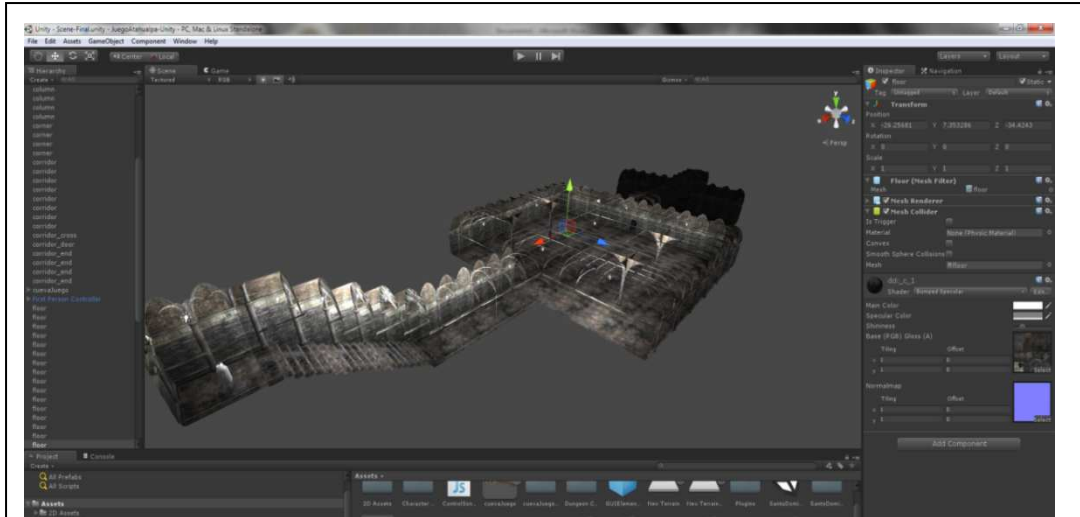


Figura 23. Texturizado interno templo  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

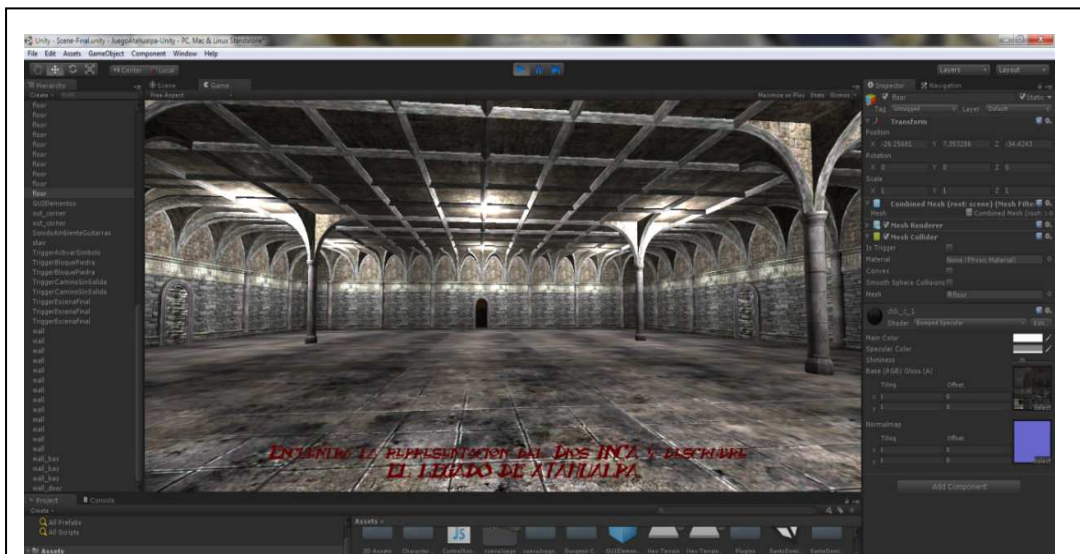


Figura 24. Texturizado interno templo e iluminación  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

Se debe observar de una manera amplia un escenario que se realiza en exterior, en nuestro caso en el nivel de la montaña es necesario revisar cómo afecta la niebla al nivel, esto con el fin de que no se pierda visibilidad de los objetos y estructuras.

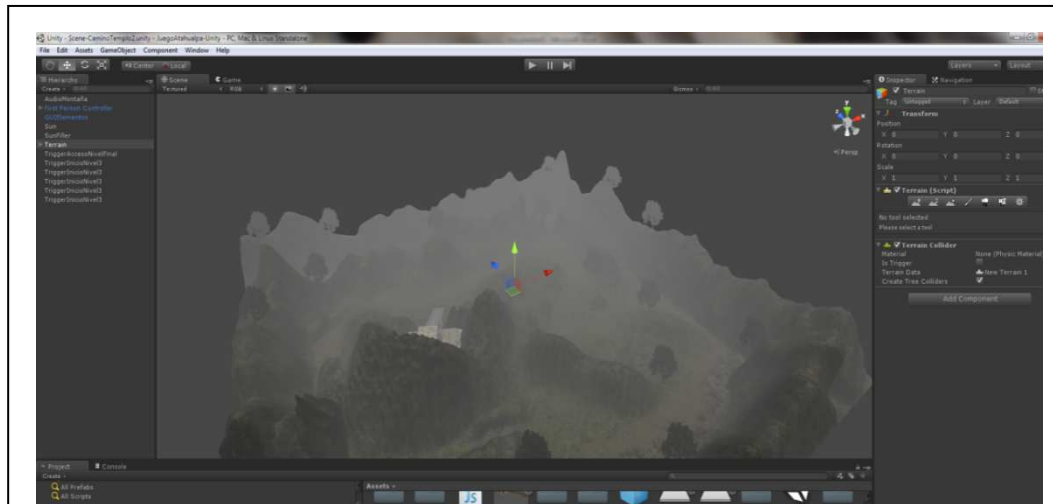


Figura 25. Texturizado exterior montaña  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

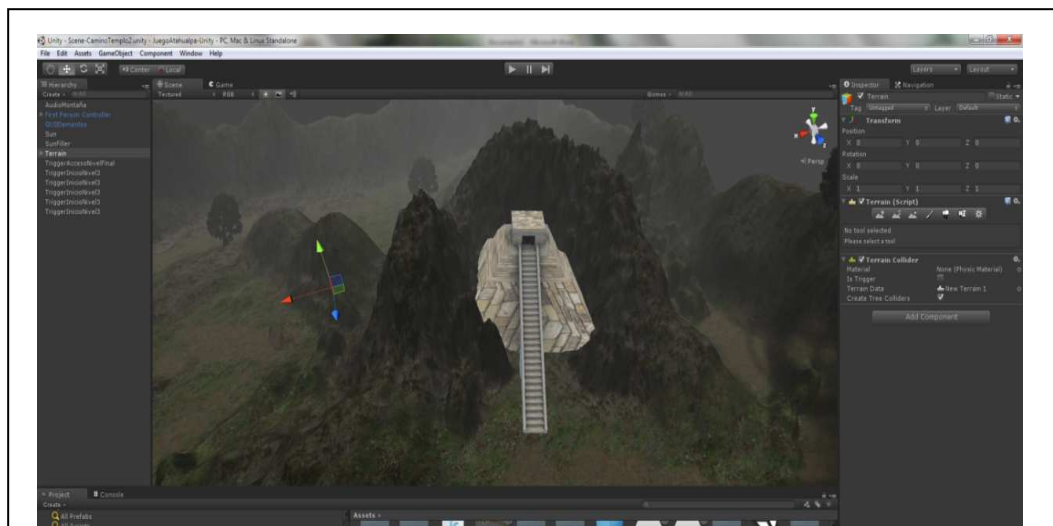


Figura 26. Texturizado exterior montaña y templo juntos  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

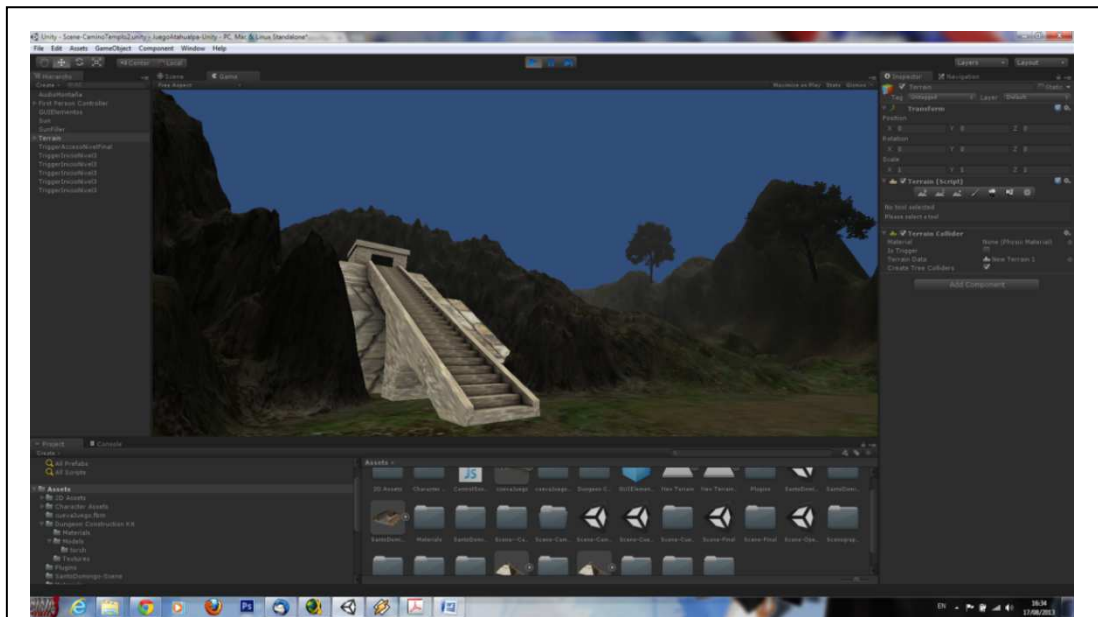


Figura 27. Texturizado exterior montaña y templo, vista frontal

Elaborado por: (Marco Alvarez & Gabriel Garzón)

#### **4.4 Audio**

Para el proceso de selección de audio en el videojuego se consideró varios factores:

- **Música:** debe apoyar el estilo, concepto del juego, para crear el ambiente en el que el usuario sienta en realidad que está dentro del videojuego. Esto representa uno de los retos más grandes para compositores ya que debe darles espacio al resto de elementos de audio que están dentro del juego; generalmente el resultado final es una mezcla de sonidos adaptables a la esencia de lo que se quiere plasmar en el juego.
- **Efectos de sonido:** estos van desde ruidos ambientales como viento, sonidos de pasos, ríos, aves, etc. junto con sonidos que el personaje o jugador realicen es decir pasos, caídas, objetos que se recojan, etc. estos sonidos le dan un mayor realismo y ayudan a definir la personalidad que el juego necesita y a la vez hace que el usuario se adentre más en el videojuego.
- **Diálogos:** hay que tomar en cuenta que para realizar esto que parece lo más sencillo se debe realizar varias pruebas tanto en casting, traducción y audiciones, para que la persona o programa que ejecute esto no haga que el juego pierda la virtud de envolver al usuario.

En nuestro caso se utilizó varios sonidos ambientales al momento de realizar el juego, estos son los sonidos del viento y agua en la montaña, así como de pisadas y música en los diferentes niveles para que mantengan atento al jugador y a lo que sucede dentro de los ambientes utilizados.

## **4.5    Diseño de la interfaz**

### **4.5.1    Exportación de modelos desde Maya a Unity.**

La transición de un modelador a un motor de juegos es algo crucial en el desarrollo, existen varias formas de poder exportar nuestros modelos 3D del ambiente de un modelador a un motor de videojuegos.

Las que más se utilizan en el desarrollo de videojuegos son:

- Exportación manual.

Se refiere a tomar todo el trabajo realizado en el modelador y exportarlo como un archivo cualquiera a Unity, el motor puede leer este tipo de archivos (obj, ma, mb) y lo convierte a un formato que pueda entenderlo, normalmente a FBX. Luego de haber completado este trabajo se debe enlazar las texturas que integraban el modelo 3D manualmente en Unity.

- Exportación a través de archivos FBX.

Un archivo FBX es un formato de archivo 3D independiente y gratuito, que proporciona acceso al contenido creado en cualquier paquete de diseño y desarrollo 3D, con esta opción se exporta el archivo FBX y Unity enlaza tanto las texturas como el modelo 3D y los organiza en el proyecto.

Aquí vale aclarar la razón por la cual existen estos dos tipos de exportación, dado que al existir la exportación a través de archivos FBX, archivos que al final son los que maneja el motor de juegos Unity.

Algunos de los problemas que se presentan al realizar la exportación a través de archivos FBX son las siguientes:

- Se puede dar el caso que al convertir el archivo FBX original a la versión que maneja Unity pueda existir conflictos.

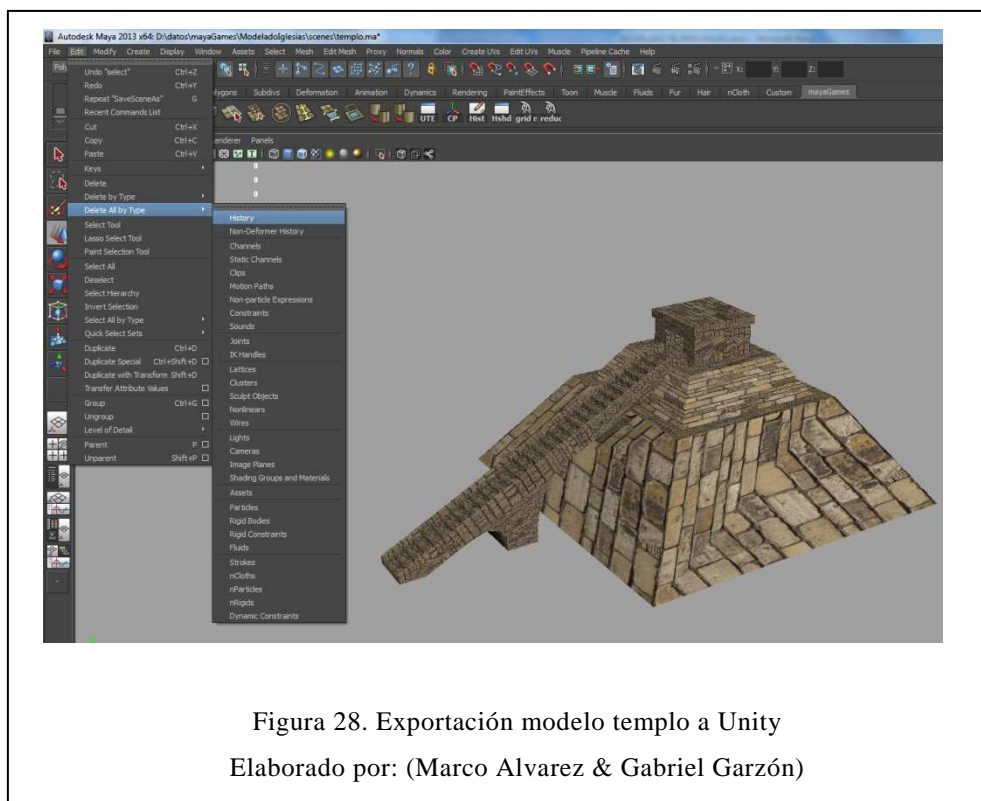
- El proceso de conversión es automático y no se tiene un control sobre él, si falla se tiene que averiguar si hay errores en el modelado original o en el sistema de conversión de Unity.
- Depende del usuario el añadir todos los elementos que conforman un modelo 3D, si falta uno, Unity no lo enlaza y esto se tendrá que realizar manualmente.
- Al exportar un modelo manualmente, se está haciendo un trabajo de backup del modelo 3D, esto sirve por ejemplo, si por algún motivo se llegará a cambiar el modelo en Unity y después de un tiempo se decide volver al original, solo se tendría que regresar al modelador y borrar lo que se tiene en el motor del juego y exportarlo nuevamente.

En el proyecto se escogió la exportación del trabajo realizado en Maya en un formato del tipo FBX. La razón es muy simple, además, de que Unity se maneja excelente con este tipo de archivos, dado que relaciona el modelo 3D con sus texturas y las organiza de tal manera que en el motor del juego se vuelve muy fácil el poder manipular dicha información, se realizaron pruebas previas con modelos 3D de ejemplo y ninguno presento algún tipo de error.

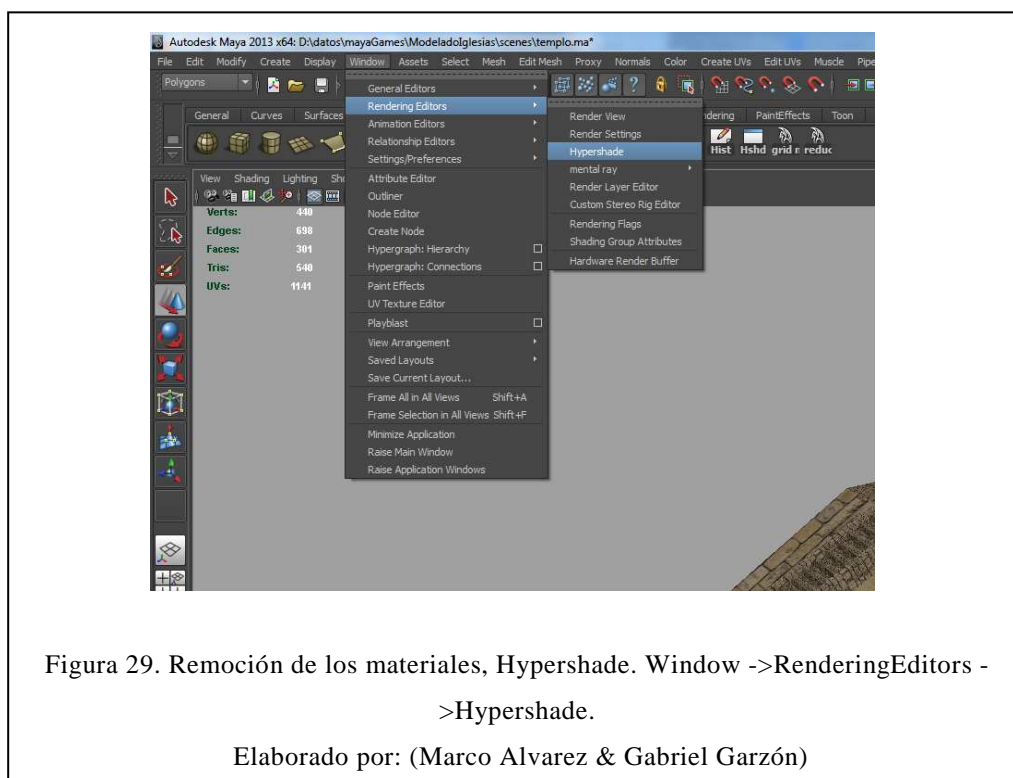
#### **4.5.2 Exportación del modelo 3D a Unity.**

Para esta tarea se debe realizar pasos previos en el modelador, en este caso Maya. Primero se debe borrar todo el historial de la escena a exportar, esto para que no exporte modelos que no pertenecen a lo que se necesita. Para ello se debe elegir en el menú de Maya: Edit ->DeleteAllByType ->History.





Es necesario realizar una remoción de materiales que afecten a la estructura del modelo a exportarse para ello se siguen los pasos de las 2 figuras a continuación.



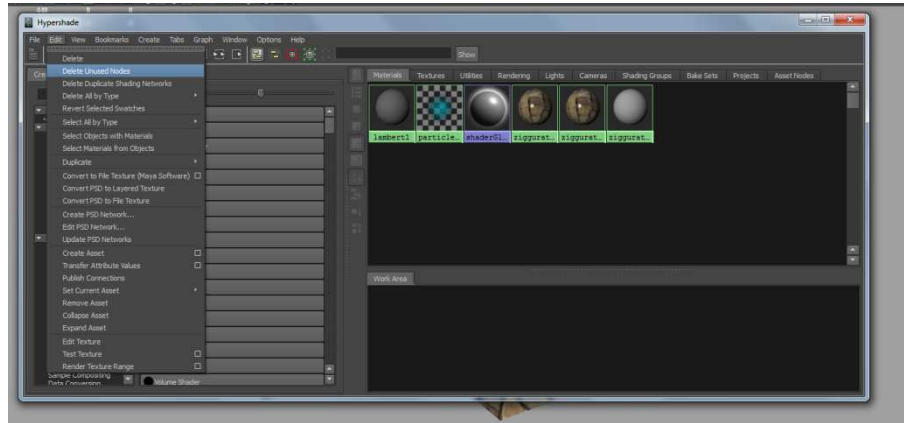


Figura 30. Remoción de los materiales que no son usados, Edit ->DeleteUnusedNodes.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Hecho esto, se debe exportar el modelo 3D para lo cual se escoge: File ->ExportAll... y se da un clic izquierdo en el cuadrado que está al lado del nombre del submenú.

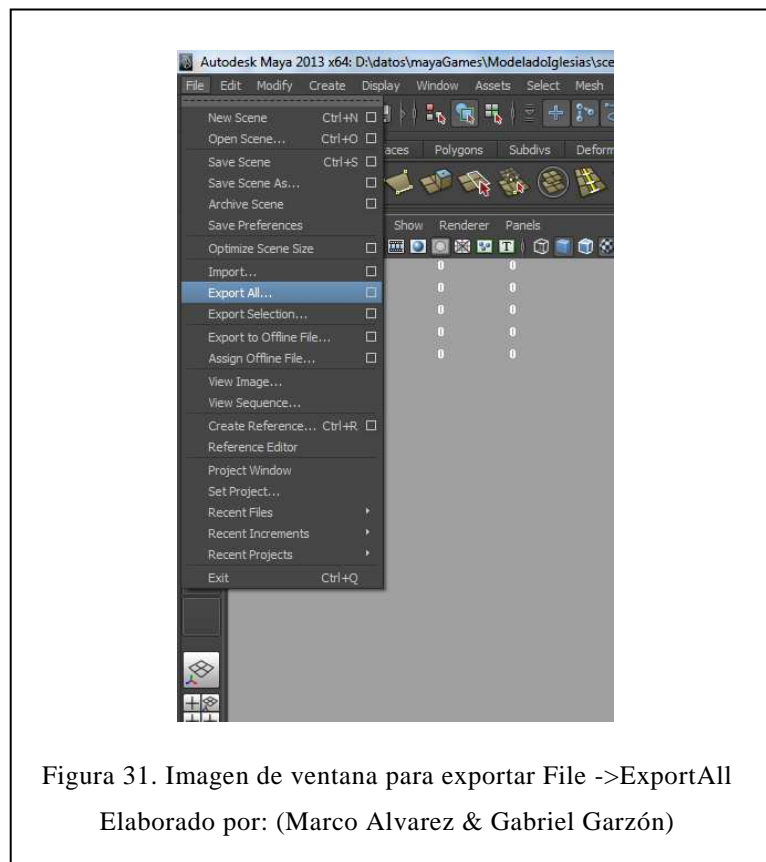


Figura 31. Imagen de ventana para exportar File ->ExportAll

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Se selecciona File Type y se da un clic en “Editpreset” esto para exportar adecuadamente y en el formato correcto.

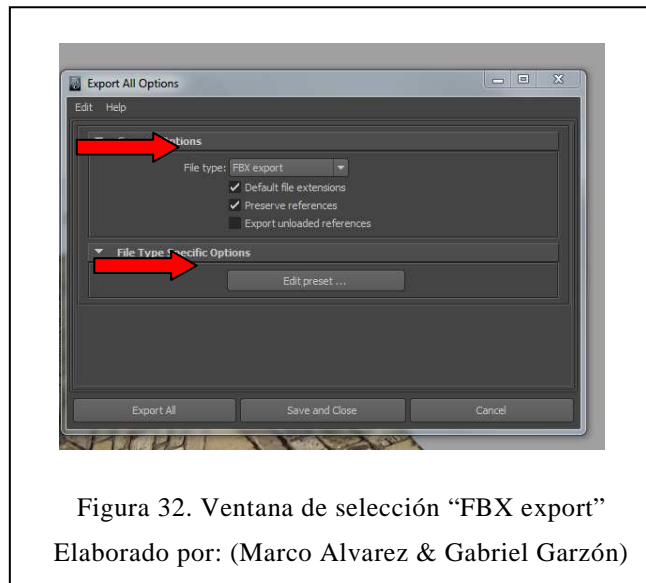


Figura 32. Ventana de selección “FBX export”  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

En la nueva ventana que se abre, se debe tener los siguientes valores, de esta manera se asegura de que tanto las texturas con el modelo serán exportados, excepto las cámaras y luces de la escena.

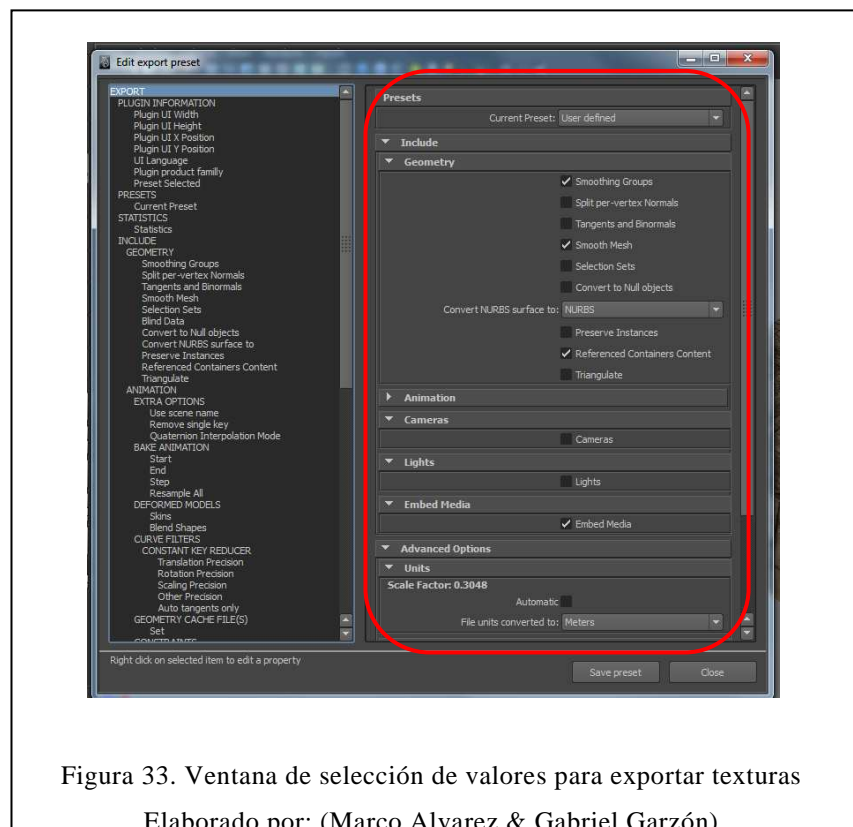


Figura 33. Ventana de selección de valores para exportar texturas  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

En la misma ventana, se ingresa con un clic en “AdvanceOptions -> FBX File-Format” y en la versión del documento se elige FBX 2010. Esto es una recomendación de los creadores de Unity.

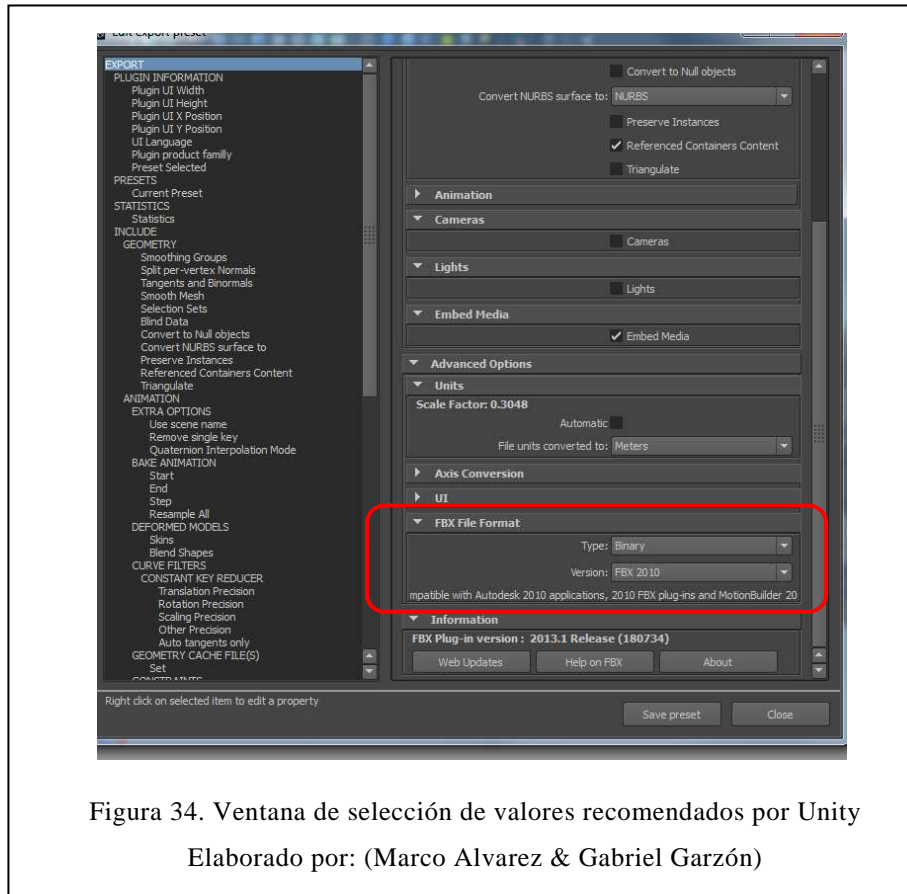


Figura 34. Ventana de selección de valores recomendados por Unity  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

Se cierra la ventana, no es necesario grabar en un nuevo preset los cambios realizados y por último se da un clic en ExportAll. Solo se tendrá que elegir el lugar en el cual se guardará la exportación.

### 4.5.3 Construcción de la Funcionalidad del juego.

#### 4.5.3.1 Estándares de la codificación.

##### 4.5.3.1.1 Declaración de variables.

Los scripts son desarrollados bajo los lenguajes Javascript y C# de Unity, los cuales se diferencian del utilizado en el desarrollo web o de aplicaciones respectivamente, estos mantienen las mismas reglas de uso como la declaración de variables o métodos, clases, etc.

Un ejemplo de la declaración de variables en Javascript se realiza de la siguiente manera.

```
//variable para activar la pista del juego en la cueva  
private var activadorPista : GameObject;  
//variable que guarda el audio para cuando se complete el r
```

En el lenguaje de desarrollo Javascript, el acceso a las variables que son declaradas es pública, es decir, no es necesario indicar mediante la palabra reservada “public”, este nivel de acceso, solo se tiene que declarar la variable de esta forma.

```
var activadorPista : GameObject;
```

Pero si la variable declarada necesita un nivel de acceso privado se debe indicar esto mediante la palabra reservada “private” antes de la palabra reservada “var”.

En el lenguaje de desarrollo C# es necesario la declaración del nivel de acceso de una variable mediante las palabras claves “public” o “private”, en este lenguaje no es posible obviar esta regla como en Javascript.

Por lo cual se decidió el uso de las palabras claves “public” y “private” en la declaración de variables como una regla principal, sin importar el lenguaje en que se esté desarrollando.

El nombre que tendrá una variable corresponderá, en la mayoría de los casos, al de la acción u evento en que será utilizada.

#### 4.5.3.1.2 Declaración de funciones.

Para la declaración de funciones se siguen los estándares o reglas de cada lenguaje.

El nombre del método corresponderá al comportamiento u acción que deba realizar al ser ejecutado. Por ejemplo, se provee la siguiente imagen en la cual se puede apreciar el código fuente de la función “ComprobarCodigoCueva” que sirve para revisar si el jugador ha realizado la elección correcta dentro de un mini juego en el proyecto.

```
function ComprobarCodigoCueva(codigoUsuario : String)
{
    if(codigoUsuario == "2143")
    {
        juegoSuperado = true;
        PasarSiguienteNivel(juegoSuperado);
        Destroy(activadorPista);
        msjJuegoSuperado.guiTexture.enabled = true;
        audioJuegoSuperado.audio.Play();
        yield WaitForSeconds(audioJuegoSuperado.audio.clip.length);
        guiMensajes.GetComponent(SantoDomingoGUITextScript).MensajeJuegoResueltoCueva();
        msjJuegoSuperado.guiTexture.enabled = false;
    }
    else
    {
        juegoSuperado = false;
        PasarSiguienteNivel(juegoSuperado);
    }
}
```

La indentación toma el valor de una unidad de tabulador.

Siempre que se requiera en una línea de código asignar o comparar elementos, siempre habrá un espacio entre variables y signos, por ejemplo:

- codigoUsuario = 10;
- codigoUsuario == 10;

Ejemplo:

```
function SumarDosValores(valor1 : int, valor2 : int)

{

    return (valor1 + valor2);

}
```

En lo posible, en los nombres de variables y métodos, se tratará de utilizar palabras acordes al uso de la variable o método y siempre tratando de que estén en español.

#### 4.5.3.2 *Script ControlSonidoPisadas.js.*

El script controla el sonido relacionado con las pisadas, cambia dependiendo de que el personaje se encuentre en un ambiente cerrado (Iglesia, Cueva, Catacumbas) o en uno abierto (Bosque).

En el proyecto se tiene la carpeta llamada *Sonidos*, en esta se guardan los diferentes archivos de audio que se utiliza a lo largo del juego, como el sonido de los pilares en la iglesia, el audio que ambienta la escena (sonido de fondo), o las pisadas del personaje.

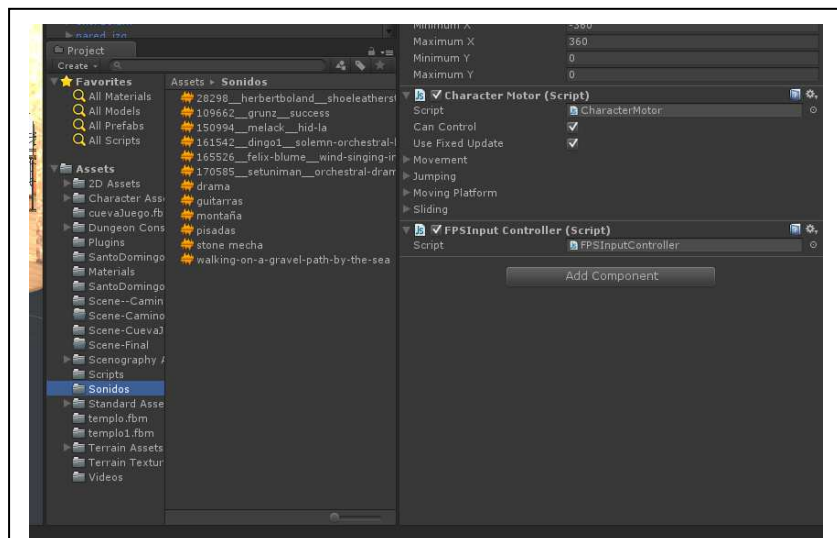


Figura 35. Carpeta Sonido en Unity.  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

Los audios que representan las pisadas del personaje solo pueden ser escuchadas al momento que se presiona los botones de dirección, es decir, los botones A, S, D, W o las flechas.

Dado que se debe de comprobar en cualquier momento si estas teclas están presionadas, la función Update contiene las tareas para esta acción.

```

6
7  function Update () {
8
9      //Revisamos si los botones para moverse Vertical u Horizontalmente estan presionados
10     if (Input.GetButton("Vertical") || Input.GetButton("Horizontal"))
11     {
12         //Prendemos el audio de pisadas
13         audio.enabled = true;
14     }
15     else
16     {
17         //Se apaga el audio, deja de oír las pisadas
18         audio.enabled = false;
19     }
20
21 }

```

#### 4.5.3.3 Script FuncionalidadCuevaScript.js.

Este script controla todo el funcionamiento dentro de la escena en la cueva, debajo de la iglesia. Muestra los diferentes mensajes de ayuda y determina si el jugador ha superado la prueba en la cueva para pasar al siguiente nivel.

Primero se declaran las diferentes variables que se utilizan en el script como se muestra a continuación.

```

/*****
Declaracion de SerializePrivateVariables
*****/

//variable para activar la pista del juego en la cueva
private var activadorPista : GameObject;
//variable que guarda el audio para cuando se complete el primer nivel
private var audioJuegoSuperado : GameObject;
//variables que guardan el audio para cuando se activen las imagenes del juego
private var audioMecanismo1 : GameObject;
private var audioMecanismo2 : GameObject;
private var audioMecanismo3 : GameObject;
private var audioMecanismo4 : GameObject;
//variable que guarda el texto para ayudar al gamer en el juego
private var ayudaTexto : GameObject;
//variable que guarda un objeto del juego, el modelo del Dios Inti
private var modeloCaraInti : GameObject;
//variables contadores, para cada imagen nos diran si esta o no activada una imagen
private var cont1 : int;
private var cont2 : int;
private var cont3 : int;
private var cont4 : int;
private var cont5 : int;

```



```

//variable que mostrara los mensajes en pantalla
private var guiMensajes : GameObject;
//variables que guardan las imagenes del juego en la cueva para mostrarla en pantalla
private var imgAyuda1 : GameObject;
private var imgAyuda2 : GameObject;
private var imgAyuda3 : GameObject;
private var imgAyuda4 : GameObject;
//variable que dira si el juego ha sido superado o no
private var juegoSuperado : boolean;
//variables que guardan objetos dentro del juego, en este caso las representaciones
//en la cueva
private var mecanismo1 : GameObject;
private var mecanismo2 : GameObject;
private var mecanismo3 : GameObject;
private var mecanismo4 : GameObject;
//variable que guarda el mensaje de juego superado
private var msjJuegoSuperado : GameObject;
//variable que guardara la secuencia que realiza el usuario al resolver el juego de la cueva
private var secuenciaUsuario : String;
//variable que guardara la vinculacion hacia el siguiente nivel
private var sigNivel : GameObject;
//variable que ayuda a saber que tipo de mensaje mostrar al jugador en pantalla para
//resolver el juego de la cueva
private var status : boolean;

```

Continuando, se inicializan algunas variables.

```

51 //Inicializamos los contadores
52 cont1 = 0;
53 cont2 = 0;
54 cont3 = 0;
55 cont4 = 0;
56 cont5 = 0;
57 //Al comenzar el juego esta variable debe tener valor false
58 juegoSuperado = false;
59 //Inicializamos la secuencia del usuario en vacio
60 secuenciaUsuario = "";

```

Luego, se encuentra la función especial “Awake()”, esta función se ejecuta primero, después de cargarse la escena del juego, se utiliza dicha función dado que los métodos que contiene generan algo de sobrecarga de memoria en la plataforma en la cual se ejecuta el juego.

Entre éstas se encuentra la función “GameObject.GameFind()”, se recomienda por rendimiento que esta función sea ejecutada una vez y no cada segundo o frame por segundo. (UnityDocumentation, 2012)

```

//Esta funcion actua al cargar la escena del juego y solo una vez
function Awake ()
{
    //se inicializan las diferentes variables con objetos activos en la escena de la cueva
    //GameObject.Find es recomendable solo utilizarlo una vez por objeto.
    activadorPista = GameObject.Find("TriggerActivadorJuegoCueva");
    audioJuegoSuperado = GameObject.Find("SonidoJuegoSuperado");
    audioMecanismo1 = GameObject.Find("SonidoMecanismo1");
    audioMecanismo2 = GameObject.Find("SonidoMecanismo2");
    audioMecanismo3 = GameObject.Find("SonidoMecanismo3");
    audioMecanismo4 = GameObject.Find("SonidoMecanismo4");
    ayudaTexto = GameObject.Find("GUIAyudaTexto");
    guiMensajes = GameObject.Find("GUIElementos");
    //cargamos las imagenes que nos darán pistas para resolver el puzzle
    imgAyuda1 = GameObject.Find("GUIAyuda1");
    imgAyuda2 = GameObject.Find("GUIAyuda2");
    imgAyuda3 = GameObject.Find("GUIAyuda3");
    imgAyuda4 = GameObject.Find("GUIAyuda4");
    //el mensaje que se muestra en pantalla es una imagen por lo cual debemos buscarla con GameObject.Find
    msjJuegoSuperado = GameObject.Find("GUILicitudes");
    //buscamos y cargamos los pedestales en el juego de la cueva
    mecanismo1 = GameObject.Find("piedraJuego1");
    mecanismo2 = GameObject.Find("piedraJuego2");
    mecanismo3 = GameObject.Find("piedraJuego3");
    mecanismo4 = GameObject.Find("piedraJuego4");
    //cargamos el modelo del dios inti en esta variable
    modeloCaraInti = GameObject.Find("caraIntiCueva");
    //este objeto es el link entre la cueva y el siguiente nivel
    sigNivel = GameObject.Find("TriggerSigNivel");

    //al tener un status = true solo debera mostrar en pantalla la ayuda basica del juego en la cueva
    //no mostrara la ayuda individual para cada imagen dentro de la cueva
    status = true;
}

```

Las variables que se inicializan en este método son importantes para la correcta funcionalidad del juego.

La variable “activadorPista” se inicializa con un elemento llamado “TriggerActivadorJuegoCueva” este elemento maneja los mensajes que puede mostrar la escena para que el jugador conozca que debe hacer.



Figura 36. Ejemplo de mensajes de ayuda que tiene el jugador.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Luego se cargan los diferentes archivos de audio que se maneja dentro de la escena. Las siguientes variables se inicializan con elementos que controlan otro

tipo de mensajes que aparecen en la interfaz del jugador, estos son utilizados al interactuar con el inventario en el juego.

Las variables “mecanismo” se inicializan con los elementos que representan el mini juego “La Historia del Primer Inca” y la imagen del “Dios Inti”. La variable “sigNivel” permitirá al jugador, una vez superado el mini juego, continuar al siguiente nivel.

La variable “status” indicará que tipo de ayuda mostrar al jugador, una ayuda básica del nivel o general de cada elemento del mini juego. A continuación se tiene la función “Update()”, esta función se ejecuta cada cuadro por segundo (frame per second). (UnityDocumentation, 2012)

Para explicar las siguientes partes del código primero se tiene que revisar cómo está configurada una pieza del juego en la cueva debajo de la iglesia de Santo Domingo.

En la figura 37. Se puede observar en la parte señalada los componentes que conforman el elemento seleccionado bajo la pestaña llamada “Inspector”.

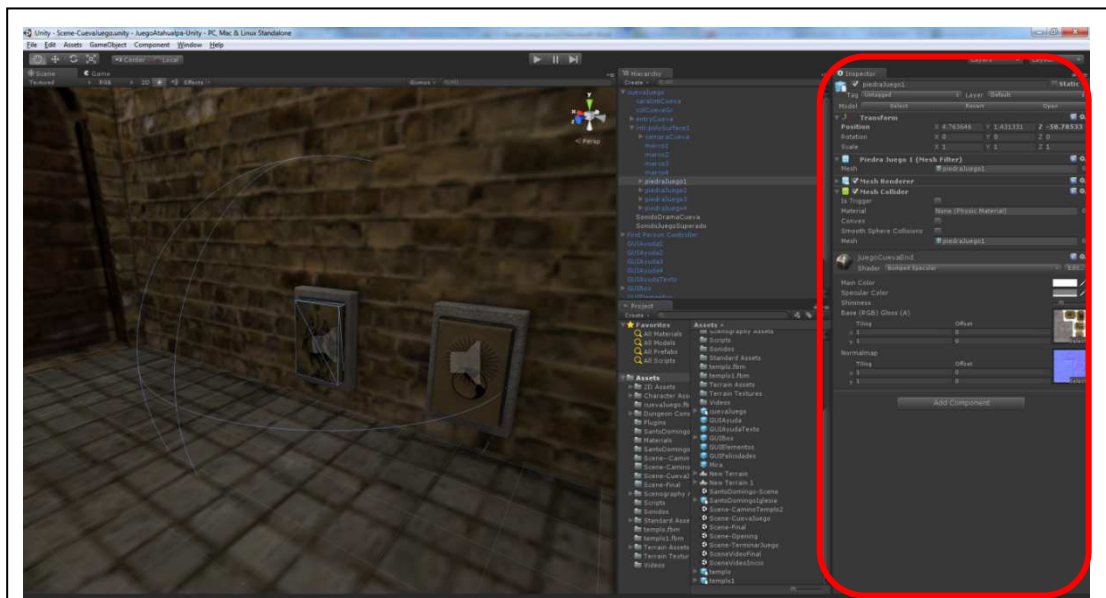


Figura 37. Elemento del juego “Historia Inca”.

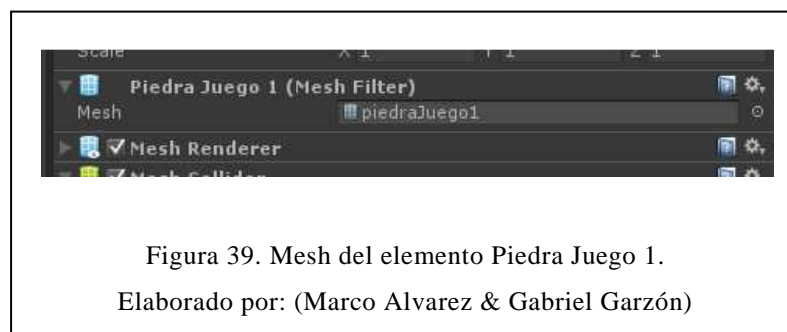
Elaborado por: (Marco Alvarez & Gabriel Garzón)

En esta pantalla se podrá apreciar los diferentes componentes que puede tener un elemento dentro de una escena (audio, texturas, scripts, animaciones, etc.).

Lo primero que se puede revisar es el componente “Transform” en el cual se puede manipular la posición, rotación o tamaño del elemento.



En la parte inferior del componente Transform se encuentra el tipo de “Mesh” (Maya) del elemento, una mesh representa al elemento dentro de la escena para que este pueda tener interacción con el personaje, normalmente una mesh tiene la forma del elemento pero puede ser más básica y tener formas de cubos, esferas, capsulas, etc. Esto se realiza para obtener un mejor rendimiento en el juego y no sobrecargar tanto la plataforma en donde se ejecute.



Lo siguiente que se puede revisar es el componente “Mesh Renderer”, este componente permite que el elemento pueda ser renderizado, que no es más que generar el elemento a partir del cálculo de iluminación que proporcione la escena

en donde se encuentre dicho elemento. Con esto se puede generar sombras, brillos u otro tipo de efectos.

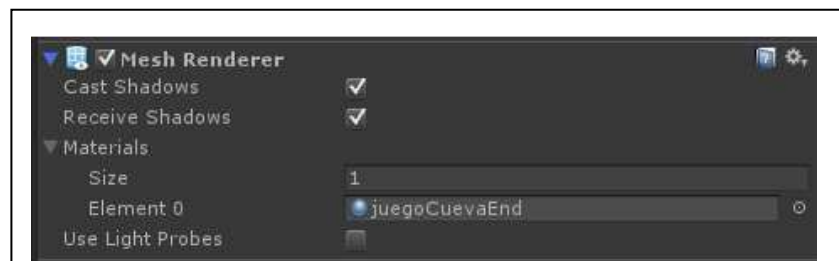


Figura 40. Componente Mesh Renderer.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

El siguiente componente es “Mesh Collider”. Este componente es importante siempre y cuando sea necesario generar colisiones con el elemento de la escena, hay diferentes motivos por el cual generar colisiones entre el personaje y algún elemento de la escena, la principal es la interactividad (presionar un interruptor, subir un ascensor, abrir una puerta, etc.).

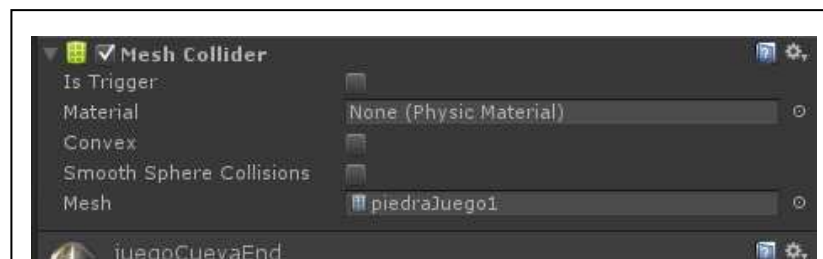
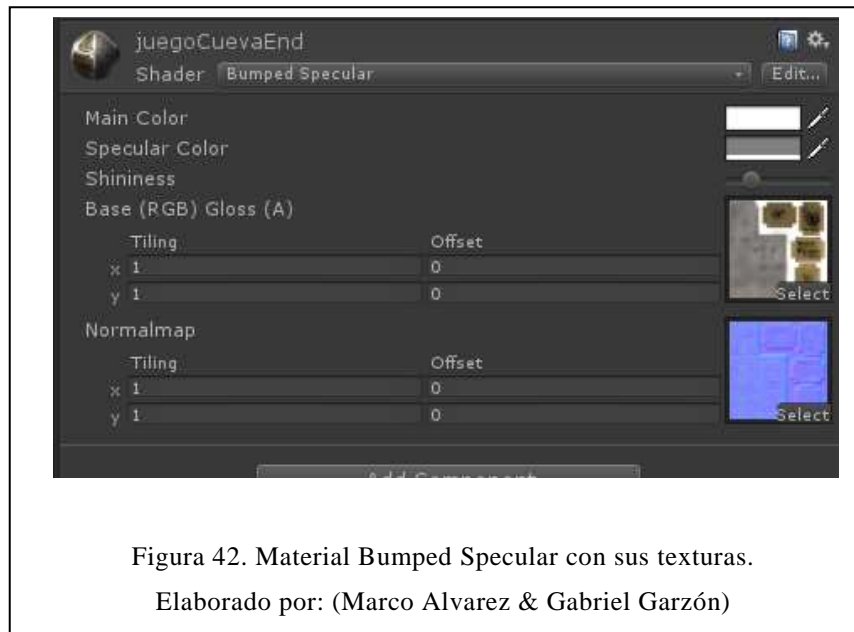


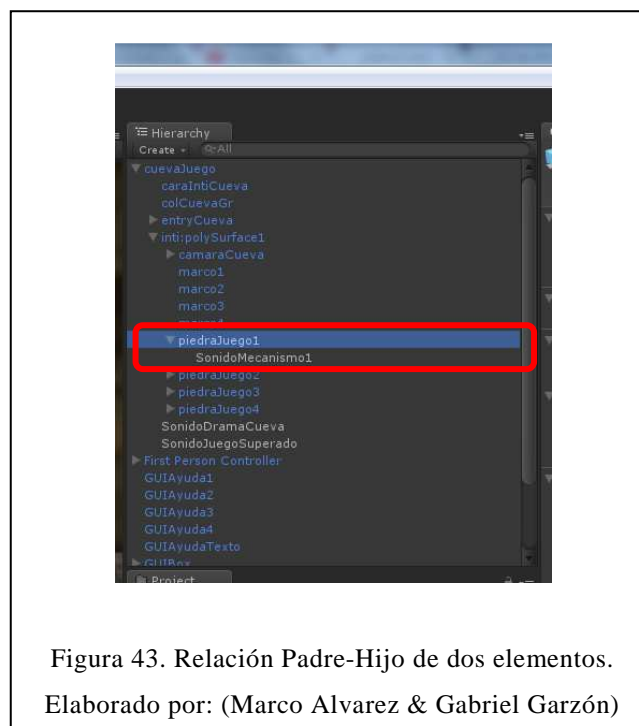
Figura 41. Componente Mesh Collider.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Por último, también se puede modificar los materiales que estén relacionados a un elemento. En este caso se puede manipular un material con dos tipos de texturas, Color y el NormalMaps.



El elemento también tiene vinculado una pista de audio, la cual puede ser modificada o editada de la siguiente manera. Un archivo o pista de audio también es considerado un elemento como los modelos 3D o el personaje dentro de una escena, por lo cual puede ser vinculada a otro elemento mediante una relación Padre-Hijo. En la pantalla “Hierarchy” se da un clic izquierdo a la flecha al lado del nombre del elemento seleccionado, mostrará los elementos que son “hijos” del elemento padre. En este caso será “SonidoMecanismo1” hijo del elemento “piedraJuego1”.



Una vez explicada esta parte se podrá entender un poco mejor las siguientes líneas de código del script.

Como se explica en párrafos anteriores, se procede a crear la función “Update()” la cual contiene el siguiente código.

---

```
//Esta funcion se ejecuta cada frame por segundo (cuadro por segundo)
function Update ()
{
    //esta variable nos indicara si estamos viendo a una imagen
    //en realidad es un rayo de luz
    var hit : RaycastHit;

    //Preguntamos si el rayo de luz a partir de la posicion del personaje y unos 10 mts(escala del juego)
    //golpea con otro objeto
    if(Physics.Raycast(transform.position, transform.forward, hit, 10))
    {
        //En la consola se dibujara una linea y nos mostrara un mensaje
        //esto es para uso interno
        Debug.DrawLine(transform.position, hit.point);
        Debug.Log(hit.collider.gameObject.name);

        //Revisamos si el objeto con el cual se topa el rayo de luz tiene por nombre piedraJuego1
        if(hit.collider.gameObject.name == "piedraJuego1")
        {
            //cambiamos el estado de status para mostrar la ayuda correspondiente a esta imagen
            status = false;

            //Preguntamos si se da un clic izquierdo
            if(Input.GetMouseButtonDown(0))
            {
                //realizaremos el movimiento de la piedra del juego
                iTween.MoveTo(hit.collider.gameObject, Vector3(4.763646,1.431331,-58.95821), 8);

                //hacemos que el audio asociado a una imagen comience a sonar
                audioMecanismo1.audio.Play();
                //añadimos el numero correspondiente a la imagen a la secuencia
                secuenciaUsuario = secuenciaUsuario + "1";

                //Si la secuencia tiene 4 elementos Comprobamos dicha secuencia para saber si la escogida es la correcta
                if(secuenciaUsuario.Length == 4)
                {
                    ComprobarCodigoCueva(secuenciaUsuario);
                }
            }
        }
    }
}
```

En esta primera parte del código se crea una variable especial del tipo Raycast, la cual se la puede representar como un rayo de luz que indicará con que elemento de la escena se quiere interactuar. Una vez creada la variable, se provee un punto desde donde se proyectará el rayo de luz, así como también su dirección y longitud mediante la función “Physics.Raycast()”. (UnityDocumentation, 2012)

---

```
//Esta funcion se ejecuta cada frame por segundo (cuadro por segundo)
function Update ()
{
    //esta variable nos indicara si estamos viendo a una imagen
    //en realidad es un rayo de luz
    var hit : RaycastHit;

    //Preguntamos si el rayo de luz a partir de la posicion del personaje y unos 10 mts(escala del juego)
    //golpea con otro objeto
    if(Physics.Raycast(transform.position, transform.forward, hit, 10))
    {

```

Dicho método retorna un valor de verdadero o falso si la variable RaycastHit se intercepta con algún elemento de la escena que tenga un collider, para ello se debe proveer los valores de:

- Punto donde nace el RaycastHit.
- La dirección que tendrá.
- El nombre de la variable RaycastHit.
- Y su tamaño, esto dependerá de la escala en la cual este creada la escena, por default en Unity esta escala esta medida en metros, 1:1, es decir una unidad de la posición o dimensión de un elemento representa 1 metro.

En Unity, la propiedad collider se refiere a que el elemento que tenga esta propiedad podrá colisionar o interactuar con el jugador o con cualquier otro elemento en la escena. (UnityDocumentation, 2012)

El rayo de luz, que sería la representación más acorde de un RaycastHit, siempre se estará proyectando hacia el frente de donde este mirando el jugador dentro de la escena en Unity, y este rayo se moverá con el jugador. Si el rayo de luz colisiona con algún elemento dentro de la escena que entre sus componentes conste la propiedad “Mesh Collider”, este devolverá información con respecto al elemento con el cual se ha topado, como el nombre que tiene y que se lo puede revisar en el apartado “Hierarchy” de Unity.

Por ende, para la realización del mini juego “Historia del Primer Inca” que toma lugar en la escena “Scene-CuevaJuego”



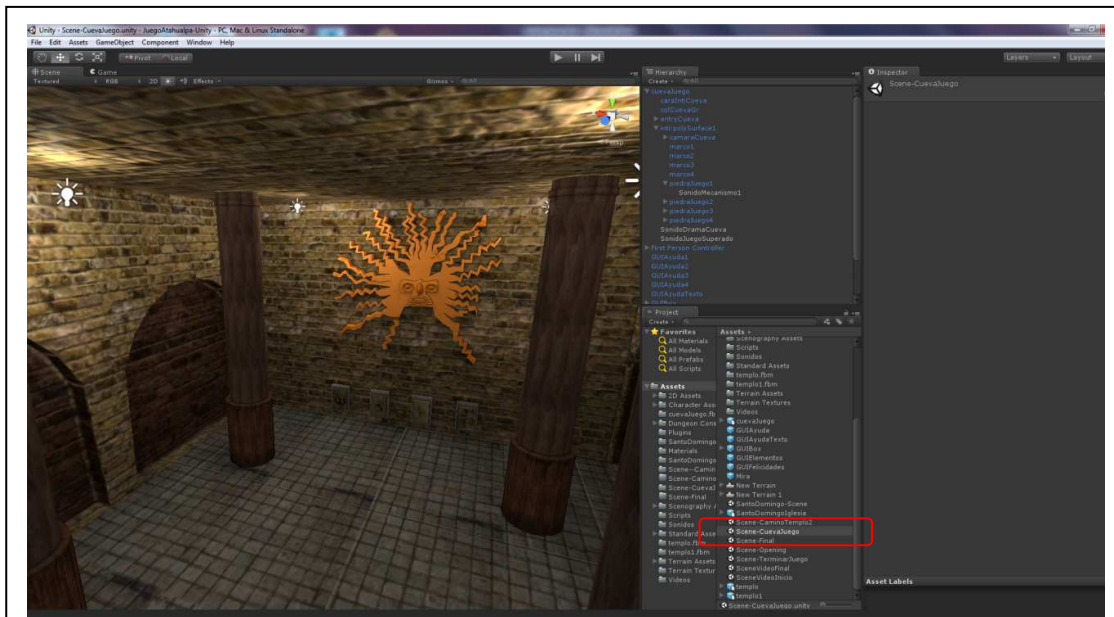


Figura 44. Escena “Scene-CuevaJuego”  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

Se necesita conocer cuando el jugador este en frente de las piedras que cuentan la “Historia del Primer Inca”.



Figura 45. Elementos del Juego “Historia del Primer Inca”.  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

Para ello es la siguiente parte del código del script.

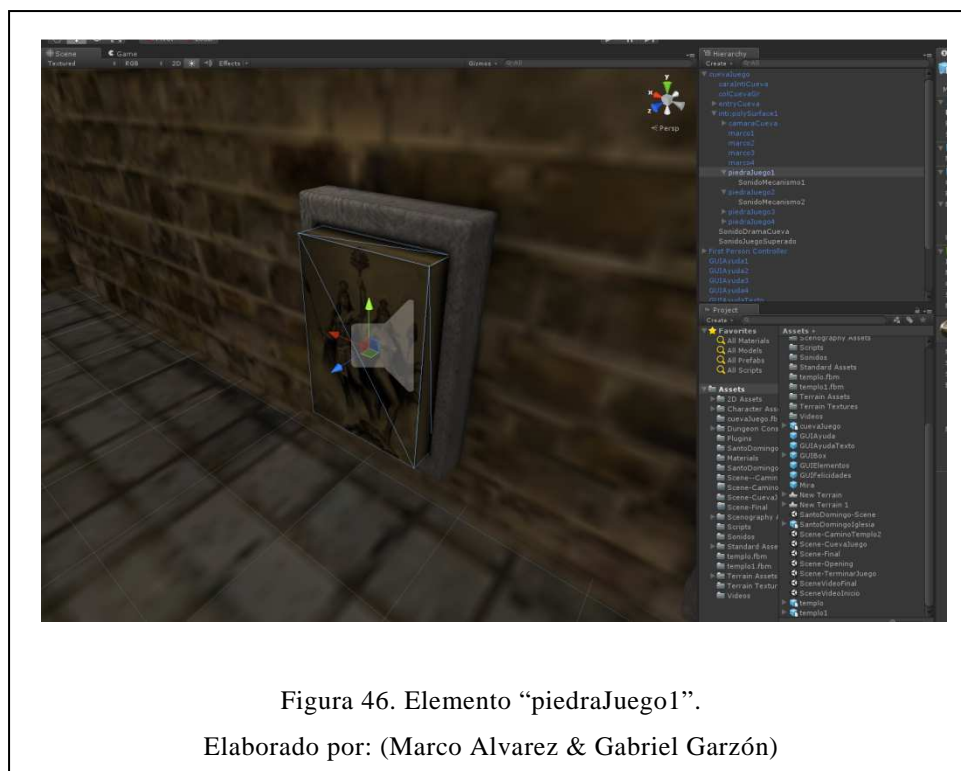
```
//Revisamos si el objeto con el cual se topa el rayo de luz tiene por nombre piedraJuego1
if(hit.collider.gameObject.name == "piedraJuego1")
{
    //cambiamos el estado de status para mostrar la ayuda correspondiente a esta imagen
    status = false;
}
```

```
//Preguntamos si se da un clic izquierdo
if(Input.GetMouseButtonDown(0))
{
    //realizaremos el movimiento de la piedra del juego
    iTween.MoveTo(hit.collider.gameObject, Vector3(4.763646,1.431331,-58.95821), 8);

    //hacemos que el audio asociado a una imagen comience a sonar
    audioMecanismo1.audio.Play();
    //añadimos el numero correspondiente a la imagen a la secuencia
    secuenciaUsuario = secuenciaUsuario + "1";

    //Si la secuencia tiene 4 elementos Comprobamos dicha secuencia para saber si la escogida es la correcta
    if(secuenciaUsuario.Length == 4)
    {
        ComprobarCodigoCueva(secuenciaUsuario);
    }
}
```

En esta parte, se condiciona si el rayo de luz ha colisionado con un elemento de la escena cuyo nombre es “piedraJuego1”, este elemento es el que se puede revisar en la siguiente imagen.



Esto quiere decir que el jugador está delante de dicho elemento, por lo cual se cambia el valor de la variable “status” la cual, al tener un valor de false o falso, indicará que se puede mostrar la ayuda con respecto al elemento “piedraJuego1”.

Luego se realiza una segunda condición, en esta se pregunta si el jugador ha realizado una acción, en este caso se refiere a si ha pulsado un botón del mouse, “Input.GetMouseButtonDown(0)”.

Hay tres diferentes acciones que se puede realizar con los botones del mouse y que se puede controlar mediante estos tres métodos:

- `Input.GetMouseButtonDown(0)` : Cuando se ha presionado el botón izquierdo del mouse.
- `Input.GetMouseButtonDown(1)` : Cuando se ha presionado el botón derecho del mouse.
- `Input.GetMouseButtonDown(2)` : Cuando se ha presionado el botón medio del mouse.

Si el jugador ha presionado el botón izquierdo en frente del elemento “piedraJuego1” se realiza la animación, en la cual este elemento, se mueve hacia la parte trasera del mismo simulando que lo se lo presionado.

Para ello se utiliza un plugin llamado “iTween”. Este plugin pertenece a un proyecto externo de Unity, pero que se puede utilizar de manera gratuita, este plugin permite simplificar las animaciones dentro de las escenas en Unity mediante código.

El proyecto tiene toda la documentación para realizar animaciones sencillas como también complicadas. (itween, 2012)

Por lo cual la siguiente línea de código realiza la animación antes explicada.

```
//Preguntamos si se da un clic izquierdo
if(Input.GetMouseButtonDown(0))
{
    //realizaremos el movimiento de la piedra del juego
    iTween.MoveTo(hit.collider.gameObject, Vector3(4.763646,1.431331,-58.95821), 8);
}
```

El método “`iTween.MoveTo`” permite mover un elemento a otra posición dentro de la escena. La nueva posición se debe establecer como un Vector de 3 dimensiones “`Vector3`” el cual determinará la posición en los ejes “X”, “Y” y

“Z”. Por último se determina el tiempo en el que se debe realizar dicho movimiento, en este caso sería de 8 segundos.

A la vez que se realiza la animación, se reproduce el audio enlazado al elemento “piedraJuego1” mediante el siguiente código.

```
//hacemos que el audio asociado a una imagen comience a sonar  
audioMecanismo1.audio.Play();  
//añadimos el numero correspondiente a la imagen a la secuencia
```

Al realizar esta animación, la variable “secuenciaUsuario” toma un valor y lo añade a cualquier otro que haya tenido antes, este valor es un número correspondiente al elemento, es decir si el elemento que interactúa con el jugador es “piedraJuego1”, la variable añade el valor de “1”, si el elemento fuera “piedraJuego2” la variable añadiría el valor de “2”.

Si esta variable tiene una longitud de cuatro dígitos, se realiza la siguiente condición.

```
//Si la secuencia tiene 4 elementos Comprobamos dicha secuencia para saber si la escogida es la correcta  
if(secuenciaUsuario.Length == 4)  
{  
    ComprobarCodigoCueva(secuenciaUsuario);  
}
```

Para poder resolver el mini juego el jugador debe tener una ayuda o guía, esto se puede lograr en base al siguiente código.

```
if(Input.GetMouseButtonDown(1))  
{  
  
    Debug.Log("Mostrar la historia de la piedra");  
    cont1++;  
    if(cont1 == 1)  
    {  
        imgAyuda1.guiTexture.enabled = true;  
        imgAyuda2.guiTexture.enabled = false;  
        imgAyuda3.guiTexture.enabled = false;  
        imgAyuda4.guiTexture.enabled = false;  
        ayudaTexto.GetComponent(MensajesAyudaJuegoIglesiaScript).AyudaCuevaMecanismo("piedraJuego1");  
        cont2 = 0;  
        cont3 = 0;  
        cont4 = 0;  
    }  
    else  
    {  
        imgAyuda1.guiTexture.enabled = false;  
        ayudaTexto.GetComponent(MensajesAyudaJuegoIglesiaScript).DesactivarImagenAyuda();  
        cont1 = 0;  
    }  
}
```

La primera parte del código indica que esta parte se ejecutará si el jugador presiona el botón derecho del mouse y en frente de uno de los elementos que conforman el mini juego y que tienen el nombre tipo “piedraJuego1”.

El método “Debug.Log()” muestra un mensaje pero en ambiente de desarrollo, específicamente en la consola de Unity, la cual se puede observar en la siguiente imagen.

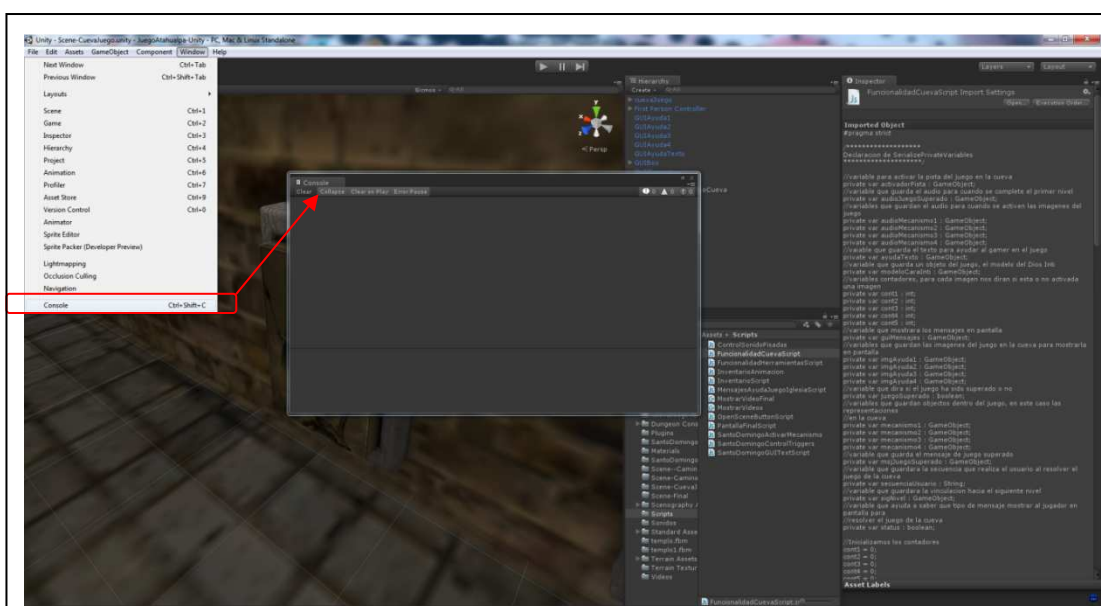


Figura 47. Consola de Unity.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

La variable “cont1” es una del tipo integer, y permite saber si el jugador ha hecho un clic para mostrar la ayuda. Para lo cual esta variable deberá tener el valor de 1, cualquier otro valor representaría que el jugador quiere ocultar los mensajes de ayuda. Cada elemento tiene su variable “cont” solo que su terminación las diferencia, ejemplo: el elemento “piedraJuego2” está relacionada con la variable “cont2”.

Si la variable “cont1” tiene el valor de uno, habilita las imágenes, dependiendo de qué elemento del mini juego este frente al jugador, en este caso al ser el elemento “piedraJuego1” la imagen que se habilita y se mostrará en pantalla será “imgAyuda1”.

Para que pueda mostrarse en pantalla la imagen cargada en la variable “imgAyuda1” se debe llamar a los métodos “guiTexture.enabled” de la variable y pasarle el valor de “true”. Es decir que esta imagen ya se encuentra en pantalla pero oculta, al pasarle el valor de “true” la imagen deja de estar oculta para mostrarse en pantalla, al mismo tiempo se muestra un mensaje debajo de dicha imagen en la pantalla, este mensaje está guardado en otro script de nombre “MensajesAyudaJuegoIglesiaScript” en el método “AyudaCuevaMecanismo(elemento : String)”.



Este código es idéntico para cada elemento, lo único que cambia es el nombre del elemento con el cual la variable “RaycastHit” se intercepta.

Se debe recordar que existen cuatro piedras que constan para el mini juego, es decir el jugador tiene que interactuar con cuatro elementos en total, al haber interactuado con ellos se debe realizar o comprobar si la secuencia que escogió el jugador es correcta o no, esto lo realiza el método “ComprobarCodigoCueva(secuencia : String)”.

```

function ComprobarCodigoCueva(codigoUsuario : String)
{
    if(codigoUsuario == "2143")
    {
        juegoSuperado = true;
        PasarSiguienteNivel(juegoSuperado);
        Destroy(activadorPista);
        msjJuegoSuperado.guiTexture.enabled = true;
        audioJuegoSuperado.audio.Play();
        yield WaitForSeconds(audioJuegoSuperado.audio.clip.length);
        guiMensajes.GetComponent(SantoDomingoGUITextScript).MensajeJuegoResueltoCueva();
        msjJuegoSuperado.guiTexture.enabled = false;
    }
    else
    {
        juegoSuperado = false;
        PasarSiguienteNivel(juegoSuperado);
    }
}

```

Este método recibe una variable del tipo String o cadena de caracteres y lo único que hace es revisar si esta secuencia es igual al valor “2143”.

Si es así la variable “juegoSuperado” toma un valor de true o verdadero y esta es tomada por otro método llamado “PasarSiguienteNivel(codigo : boolean)”. Este método se explicará más adelante.

Luego se destruye el elemento “activadorPista” para que el jugador, si por algún motivo, quiere revisar la escena una vez superado el juego, no se muestren los textos de ayuda básica que tiene el nivel.

Luego la variable “msjJuegoSuperado” es habilitada para que aparezca en pantalla, esta tiene cargada una imagen la cual sirve para indicar al jugador que el mini juego ha sido superado y puede acceder al siguiente nivel, al mismo tiempo se reproduce el archivo de audio guardado en la variable “audioJuegoSuperado”.

La variable “guiMensajes” llama a un método que consta en un script externo llamado “SantoDomingoGUITextScript” el cual es “MensajeJuegoResueltoCueva()”, el cual muestra en pantalla un mensaje que diga al jugador del recoger la imagen del “Dios Inti”.

Por último se deshabilita la variable “msjJuegoSuperado”.

La siguiente función a revisar dentro de este script es “ReactivarMecanismoCueva()”. Este método se llama si el jugador seleccionó una secuencia incorrecta en el mini juego de la cueva y realiza una animación en la cual los elementos de dicho mini juego regresan a su posición inicial.

```
function ReactivarMecanismoCueva()
{
    secuenciaUsuario = "";
    iTween.MoveTo(mecanismo1, Vector3(4.763646, 1.431331, -58.78533), 8);
    iTween.MoveTo(mecanismo2, Vector3(1.899994, 1.431331, -58.78533), 8);
    iTween.MoveTo(mecanismo3, Vector3(-0.9636572, 1.431331, -58.78533), 8);
    iTween.MoveTo(mecanismo4, Vector3(-3.827309, 1.431331, -58.78533), 8);
}
```

Por último se tiene la función “PasarSiguienteNivel(codigo : boolean)”, esta función permite continuar a la siguiente escena o nivel.

```
function PasarSiguienteNivel(codigo : boolean)
{
    if(codigo)
    {
        //haremos la animacion del modelo de INTI
        //primero lo llevaremos al centro de la habitacion
        iTween.MoveTo(modeloCaraInti, Vector3(-0.6890677, 8.461986, -46.45307), 20);

        //luego lo haremos más pequeño
        iTween.ScaleTo(modeloCaraInti, Vector3(0.594009, 0.594009, 0.594009), 20);

        //por ultimo lo rotaremos hacia el personaje y lo pondremos a ras del piso
        iTween.MoveTo(modeloCaraInti, Vector3(0.02364728, 1.364673, -46.45307), 20);
        iTween.RotateTo(modeloCaraInti, Vector3(0, -180, 0), 20);

        iTween.MoveTo(sigNivel, Vector3(0.02364728, 1.364673, -46.45307), 10);
    }
    else
    {
        ReactivarMecanismoCueva();
        guiMensajes.GetComponent(SantoDomingoGUITextScript).MensajeJuegoNoResueltoCueva();
    }
}
```

Se debe revisar primero el código del jugador que no es más que una variable que diga si el mini juego fue superado o no. Si la variable “código” tiene un valor de “true” se realizará una animación de la imagen del “Dios Inti” que se encuentra en la escena, la que hará que esta imagen se haga más pequeña y quede al centro de la sala.

Si el valor de “código” es “false” se llamará al método “ReactivarMecanismoCueva()” y se mostrará un mensaje en el cual se informará al jugador que la secuencia ingresada es incorrecta.





Figura 49. Mensaje de error en la secuencia seleccionada.

Elaborado por: (Marco Alvarez & Gabriel Garzón)



Figura 50. Imagen del Dios Inti después de la animación al completar el mini juego “Historia del Primer Inca”.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

#### 4.5.3.4 Script *FuncionalidadHerramientasScript.js*.

Este script funciona para la escena en el interior de la iglesia de Santo Domingo, la escena en exterior y que parece un terreno largo y ancho al igual en la escena final en el juego que se desarrolla en un templo. En la primera llamada “SantoDomingo-Scene” consta de un mini juego en el cual se observa pilares con marcas en ellos, estas marcas corresponden a diferentes etapas del sol, el

mini juego consiste en seleccionar en orden los pilares para formar un ciclo solar.



El script toma la misma lógica que se pudo conocer en la explicación referente al script “FuncionalidadCuevaScript.js”, en la cual se utiliza las bondades de una variable del tipo “RaycastHit” así como también del manejo de imágenes y textos para la ayuda del jugador en la resolución del mini juego.

Primeramente se declaró las variables a utilizar en este script, que como se recuerda también se utiliza en otra escena además, de la que corresponde a la iglesia de Santo Domingo.

```
private var secuenciaColumnas : String;
private var audioJuegoSuperado : GameObject;
private var audioPilar1 : GameObject;
private var audioPilar2 : GameObject;
private var audioPilar3 : GameObject;|
private var audioPilar4 : GameObject;
private var puertaSecreta : GameObject;
private var juegoSuperado : boolean;
private var activadorJuego : GameObject;
private var mecanismo : GameObject;
private var ayudaTexto : GameObject;
private var ayudaTextoJuego : GameObject;
private var ayudaImg : GameObject;
private var imgJuegoSuperado : GameObject;
private var inventario : GameObject;
private var cont : int;
public var templo : boolean;
public var terreno : boolean;
```

```

juegoSuperado = false;
secuenciaColumnas = "";

function Awake()
{
    activadorJuego = GameObject.Find("TriggerActivadorMecanismo");
    audioJuegoSuperado = GameObject.Find("sonidoJuegoSuperado");
    audioPilar1 = GameObject.Find("SonidoPilar1");
    audioPilar2 = GameObject.Find("SonidoPilar2");
    audioPilar3 = GameObject.Find("SonidoPilar3");
    audioPilar4 = GameObject.Find("SonidoPilar4");
    ayudaImg = GameObject.Find("GUIAyuda");
    ayudaTexto = GameObject.Find("GUIElementos");
    ayudaTextoJuego = GameObject.Find("GUIAyudaTexto");
    imgJuegoSuperado = GameObject.Find("GUIFelicitades");
    inventario = GameObject.Find("InventarioInti");
    mecanismo = GameObject.Find("Boton");
    puertaSecreta = GameObject.Find("entradaSecreta");
    cont = 0;
}

```

La función “Update()” es similar a la vista en el script anterior, solo que en esta se debe cambiar los nombres de los elementos que participan en las diferentes escenas.

```

function Update ()
{
    var hit : RaycastHit;

    if(Physics.Raycast (transform.position, transform.forward, hit, 10) )
    {
        Debug.DrawLine(transform.position, hit.point);
        Debug.Log(hit.collider.gameObject.name);

        if(hit.collider.gameObject.name == "caraInti")
        {
            var imgInti = inventario.GetComponent(InventarioScript).RetornarImgInti();

            if(Input.GetMouseButtonDown(0))
            {
                if(imgInti)
                {
                    ayudaTexto.GetComponent(SantoDomingoUITextScript).IntiFinal(imgInti);
                    Application.LoadLevel("SceneVideoFinal");
                    Debug.Log("Debe mostrar video final " + imgInti);
                }
                else
                {
                    ayudaTexto.GetComponent(SantoDomingoUITextScript).IntiFinal(imgInti);
                    Debug.Log("Debes sacar la imagen de inti " + imgInti);
                }
            }
        }

        if(hit.collider.gameObject.name == "Pilar1")
        {
            Debug.Log("Se debe animar pilar 1");

            if(Input.GetMouseButtonDown(0)) //El 0 significa que debemos presionar el boton izquierdo del mouse
            {
                iTween.MoveTo(hit.collider.gameObject, Vector3(0,0,0), 7);
                secuenciaColumnas=secuenciaColumnas+"1";
                audioPilar1.audio.Play();
                Debug.Log(secuenciaColumnas);

                if(secuenciaColumnas.Length == 4)
                {
                    CompararSecuenciaPilares(secuenciaColumnas);
                }
            }
        }
    }
}

```

Como se puede observar en el código anterior, los nombres de los elementos cambian, los que se utiliza en la escena de la iglesia de Santo Domingo tienen el nombre de “Pilar1” hasta “Pilar4”.

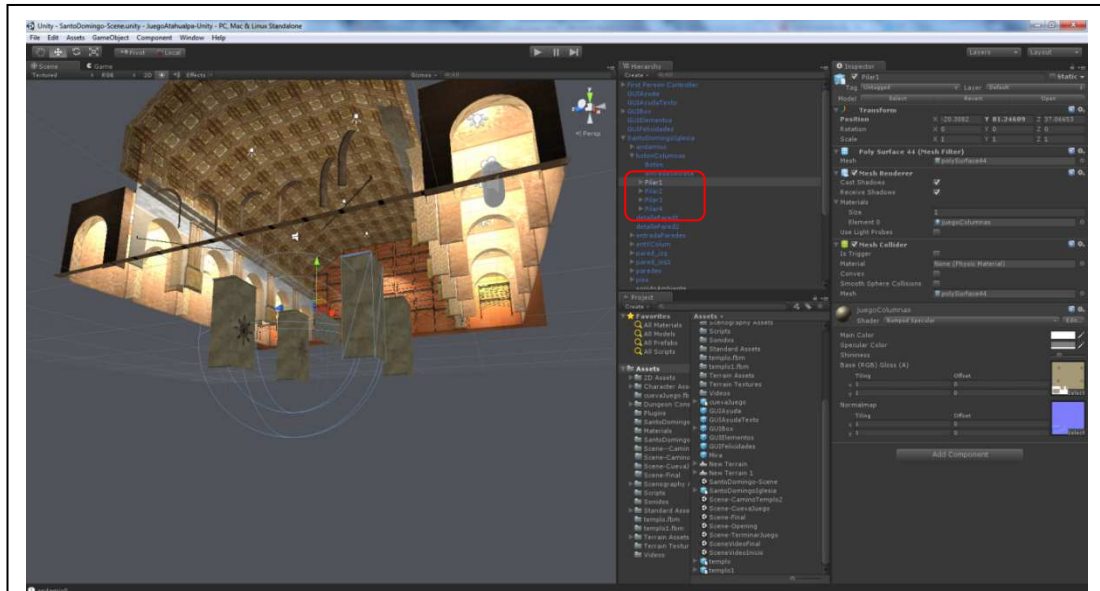


Figura 52. Nombres de los elementos que participan en el mini juego dentro de la iglesia de Santo Domingo.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Dentro de la función “Update” se puede observar una parte especial en la cual consta de dos condicionales “if” los cuales toman las variables “templo” y “terreno” respectivamente. Estas variables, al ser públicas, pueden cambiar su valor desde la misma interfaz de Unity y a determinar en qué escena deben presentar un tipo de ayuda.

En las siguientes imágenes se puedes observar los estados de estas variables en las diferentes escenas en la que este script es utilizado.

La primera imagen corresponde a la escena en el interior de la iglesia de Santo Domingo, por lo cual estas dos variables deben tener el valor de false. Para ello en la interfaz de Unity, se selecciona el elemento que tiene el script y que aparece en el área “Hierarchy”, en este caso es “Main Camera” y en la zona del “Inspector” se busca el script “Funcionalidad Herramienta Script” y se puede

observar que existen dos nombres con un checkbox a su lado, esta es la forma en que representa Unity las variables booleanas que pueden ser manipuladas (es decir que son públicas), si están seleccionadas quiere decir que estas variables tendrán un valor de “true” caso contrario será “false”.

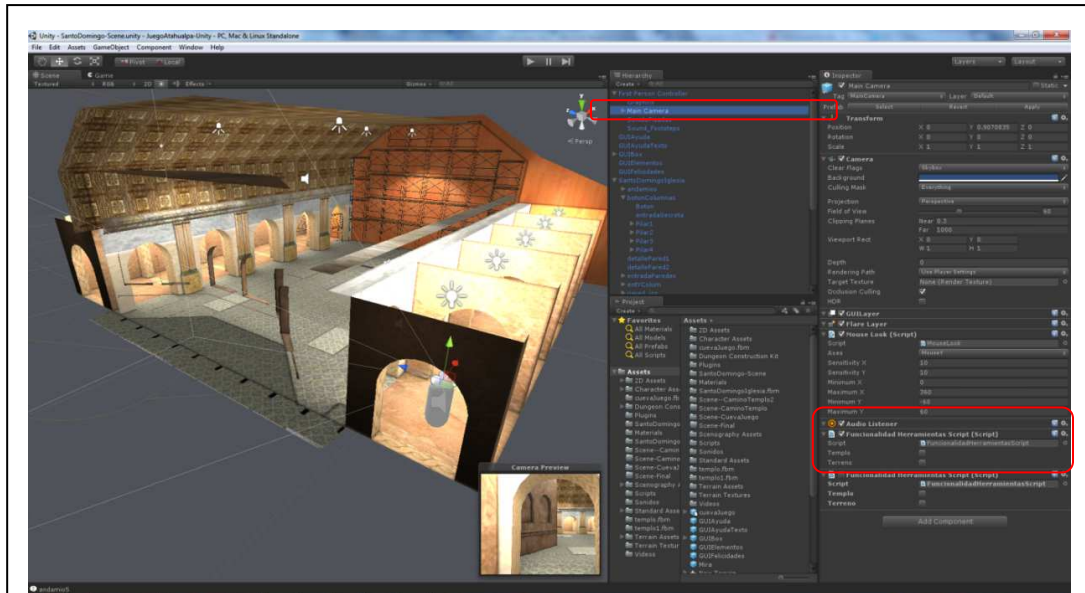


Figura 53. Representación de variables booleanas públicas en la interfaz de Unity.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Las siguientes imágenes representan las escenas “Scene-CaminoTemplo2” y “Scene-Final”, en la primera dado que el script se debe utilizar en la escena que representa un terreno, está variable tiene que tener un valor de “true” pero la variable “templo” debe tener un valor de “false”. En la segunda escena se tiene el caso contrario.



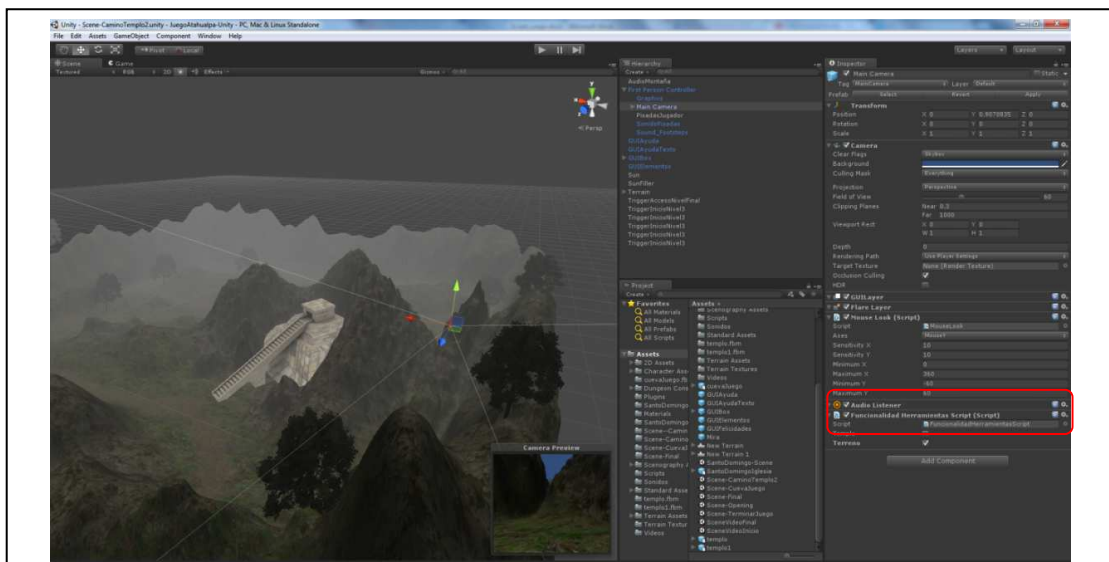


Figura 54. Estado de las variables booleanas públicas en la escena “Scene-CaminoTemplo2”.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

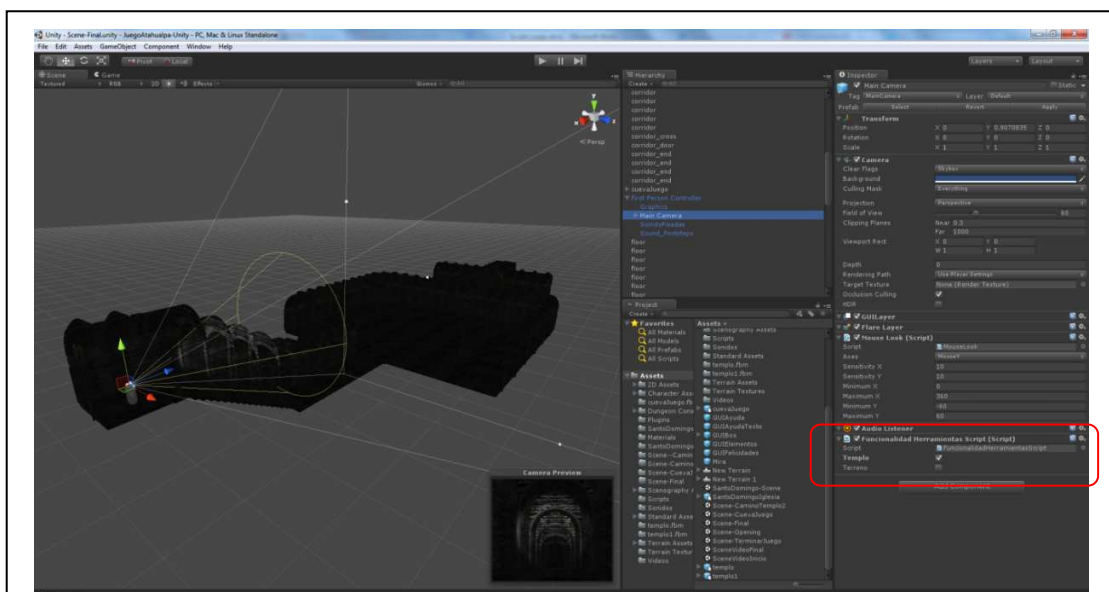


Figura 55. Estado de las variables booleanas públicas en la escena “Scene-Final”.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

La funcionalidad de estas variables se demuestra en el siguiente código, incluido en la función “Update()”, dependiendo del estado de estas variables se desplegará las ayudas básicas al jugador, dependiendo de la escena en que se encuentre.

```

if(Input.GetMouseButtonDown(1)) //1 Significa q ha presionado el boton derecho del mouse
{
    cont++;
    if(templo)
    {
        if(cont == 1)
        {
            ayudaImg.guiTexture.enabled = true;
            ayudaTextoJuego.GetComponent(MensajesAyudaJuegoIglesiaScript).ActivarImagenTemplo();
        }
        else
        {
            ayudaImg.guiTexture.enabled = false;
            ayudaTextoJuego.GetComponent(MensajesAyudaJuegoIglesiaScript).DesactivarImagenAyuda();
            cont = 0;
        }
    }
    else
    {
        if(terreno)
        {
            if(cont == 1)
            {
                ayudaImg.guiTexture.enabled = true;
                ayudaTextoJuego.GetComponent(MensajesAyudaJuegoIglesiaScript).ActivarImagenTerreno();
            }
            else
            {
                ayudaImg.guiTexture.enabled = false;
                ayudaTextoJuego.GetComponent(MensajesAyudaJuegoIglesiaScript).DesactivarImagenAyuda();
                cont = 0;
            }
        }
        else
        {
            if(cont == 1)
            {
                ayudaImg.guiTexture.enabled = true;
                ayudaTextoJuego.GetComponent(MensajesAyudaJuegoIglesiaScript).ActivarImagenAyuda();
            }
            else
            {
                ayudaImg.guiTexture.enabled = false;
                ayudaTextoJuego.GetComponent(MensajesAyudaJuegoIglesiaScript).DesactivarImagenAyuda();
                cont = 0;
            }
        }
    }
}

```

La función “AbrirPuertaSecreta(codigo : boolean)” realiza una animación de una parte del piso en la iglesia que simula una puerta secreta que llevará al jugador a la parte inferior de la misma.

```

function AbrirPuertaSecreta(codigo : boolean)
{
    if(codigo)
    {
        Destroy(activadorJuego);
        iTween.RotateTo(puertaSecreta,{"z" : -20, "time" : 10, "oncompletetarget" : puertaSecreta, "oncomplete" : "MostrarMensaje"});
    }
    else
    {
        secuenciaColumnas = "";
        ReactivarMecanismoJuego();
        ayudaTexto.GetComponent(SantoDomingoUITextScript).MensajeReiniciarJuegoIglesia();
        Debug.Log("Codigo Incorrecto, volver a activar el juego");
    }
}

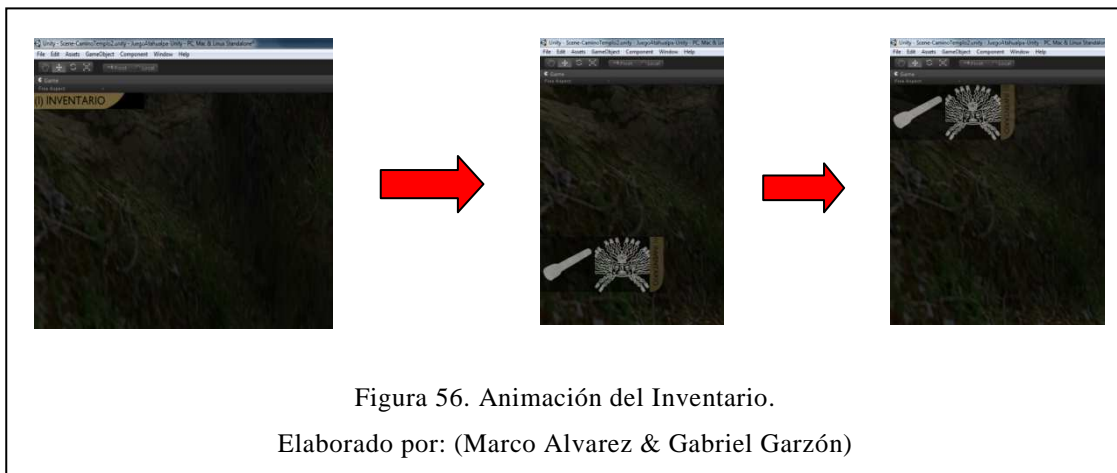
```

La función “AddHighlight(objetoEscena : GameObject)” recibe un tipo de variable “GameObject” que no es más que un elemento de la escena, y permite avivar los colores del “objetoEscena” haciéndolo más llamativo. Esto se lo realiza en la escena final del juego.

```
function AddHighlight(objetoEscena : GameObject)
{
    objetoEscena.renderer.material.color.r = objetoEscena.renderer.material.color.r*5;
    objetoEscena.renderer.material.color.g = objetoEscena.renderer.material.color.g*5;
    objetoEscena.renderer.material.color.b = objetoEscena.renderer.material.color.b*5;
    yield WaitForSeconds(0.1);
    objetoEscena.renderer.material.color.r = objetoEscena.renderer.material.color.r/5;
    objetoEscena.renderer.material.color.g = objetoEscena.renderer.material.color.g/5;
    objetoEscena.renderer.material.color.b = objetoEscena.renderer.material.color.b/5;
}
```

#### 4.5.3.5 *Script InventarioAnimacion.js.*

Este script realiza la animación del inventario dando la sensación al jugador de que está manipulando una interfaz touch.



Para ello se crean las diferentes variables y se las inicializa en la función “Start()” esta función es especial, actúa después de la función “Awake()”, si es que existiera, cuando la escena ha sido cargada totalmente.

En esta función se carga en diferentes variables los elementos que conforman el inventario y que aparecen en la pantalla del jugador, estos también son considerados elementos de la escena y tienen que estar presentes en la zona llamada “Hierarchy” en Unity. Todo lo que aparece en el inventario son imágenes.



```

private var inventarioBoton : GameObject;
private var inventarioBotonHor : GameObject;
private var camaraPrincipal : GameObject;
private var personajePrincipal : GameObject;
private var mostrarInventario : boolean;

function Start()
{
    inventarioBoton = GameObject.Find("InventarioBoton");
    inventarioBotonHor = GameObject.Find("InventarioHorizontal");
    camaraPrincipal = GameObject.Find("Main Camera");
    personajePrincipal = GameObject.Find("First Person Controller");
    mostrarInventario = false;
    Screen.lockCursor = true;
}

```

En la función “Update()” solo se revisa si el jugador presiona la tecla “I” en cualquier momento del juego, esto llama a otro método llamado “MostrarElementosInventario()”. Este realiza una serie de acciones.

Primero revisa el estado de la variable “mostrarInventario” que es una variable booleana, si tiene el valor de “true” realiza la animación de los botones, los cuales se desplazan desde la parte inferior-izquierda de la pantalla hasta la parte superior-izquierda de la misma. El valor de “mostrarInventario” cambia a “false” dado que si vuelve a presionar la tecla “I” debe realizar la animación para ocultar los botones.

Segundo, el cursor de la pantalla se vuelve estático, normalmente en el transcurso del juego el desplazamiento se lo realiza con las teclas:

- W ó flecha arriba: Para desplazarnos hacia adelante.
- S ó flecha abajo: Para desplazarnos hacia atrás.
- A ó flecha izquierda: Para desplazarnos hacia la izquierda.
- D ó flecha derecha: Para desplazarnos hacia la derecha.
- Movimiento del mouse hacia adelante: Para rotar nuestra perspectiva hacia arriba.
- Movimiento del mouse hacia atrás: Para rotar nuestra perspectiva hacia abajo.
- Movimiento del mouse hacia izquierda: Para rotar nuestra perspectiva hacia la izquierda.

- Movimiento del mouse hacia derecha: Para rotar nuestra perspectiva hacia la derecha.

Al hacer el cursor estático, los movimientos del mouse no surtirán efectos quedando nuestra pantalla estática y pasando a un modo como el de cualquier pantalla en un programa de computadora, en donde se puede dar un clic sobre los botones del inventario.

```
function MostrarElementosInventario()
{
    if(mostrarInventario)
    {
        iTween.MoveTo(inventarioBotonHor, Vector3(0, 1, 0), 1);
        iTween.MoveTo(inventarioBoton, Vector3(-0.5, 1, 0), 2);
        mostrarInventario = false;
        Screen.lockCursor = true;
        camaraPrincipal.GetComponent("MouseLook").enabled = true;
        personajePrincipal.GetComponent("MouseLook").enabled = true;
    }
    else
    {
        iTween.MoveTo(inventarioBotonHor, Vector3(0, 1.1, 0), 1);
        iTween.MoveTo(inventarioBoton, Vector3(0,1,0), 2);
        mostrarInventario = true;
        Screen.lockCursor = false;
        camaraPrincipal.GetComponent("MouseLook").enabled = false;
        personajePrincipal.GetComponent("MouseLook").enabled = false;
    }
}
```

#### 4.5.3.6 *Script InventarioScript.js.*

Este script controla la funcionalidad del inventario así como también los diferentes efectos y mensajes que se presentan al elegir o seleccionar algún elemento del inventario.

Primero se definen las diferentes variables que se van a utilizar y algunas de ellas se inicializan en la función “Start”.

```
.
var botonSeleccionado : Texture2D;
var botonNoSeleccionado : Texture2D;
var cueva : boolean;
var iglesia : boolean;
var templo : boolean;
var terreno : boolean;
private var imgInti : boolean;
private var objetoLinterna : GameObject;
private var inventarioTexto : GameObject;
private var cont : int;
private var cont1 : int;

function Start()
{
    cont = 0;
    cont1 = 0;
    imgInti = false;
}
```

Ya que el inventario está presente en todas las escenas del juego, es de vital importancia el poder controlar cuando los elementos son utilizables y cuando no, por ejemplo, en el inventario se tiene la imagen del Dios Inti, este solo estará disponible y se podrá utilizar una vez se haya resuelto el mini juego en la cueva debajo de la iglesia de “Santo Domingo”. Por lo que al elegirlo en alguna otra escena se muestra mensajes explicando al jugador que ese ítem aún no está disponible.

Es por esto que en la función “Awake()” se integra un condicional que pregunta si la variable “templo” es igual a “true”, esta variable tendrá ese valor siempre y cuando el jugador se encuentre en la escena final del juego.

```
function Awake()
{
    inventarioTexto = GameObject.Find("TextoInventario");
    if(templo)
    {
        objetoLinterna = GameObject.Find("Linterna");
    }
}
```

La función “RetornarImgInti” solo retorna el valor que tenga la variable “imgInti” la cual es una variable del tipo booleano.

Las funciones “OnMouseOver()” y “OnMouseExit()” funcionan con los elementos del inventario, cuando se coloca el puntero sobre cualquier ítem el jugador debe tener algún tipo de información de cual elemento quiere utilizar o quiere dar un clic sobre él. Para ello lo único que se realiza es un cambio en los colores para que sea más llamativa el ítem a seleccionar.

```
function RetornarImgInti()
{
    return imgInti;
}

function OnMouseOver()
{
    guiTexture.texture = botonSeleccionado;
    guiTexture.color = Color(0.5, 0.5, 0.5, 0.5);
}

function OnMouseExit()
{
    guiTexture.texture = botonNoSeleccionado;
    guiTexture.color = Color(0.5, 0.5, 0.5, 0.25);
}
```



La siguiente función en este script se refiere al dar un clic sobre cualquiera de los ítems del inventario “OnMouseDown()” en esta se diferencia entre los dos ítems del inventario, si se realiza un clic sobre uno de ellos esta acción genera un evento el cual devuelve el nombre del ítem. Dependiendo de la escena en la que se encuentra genera un mensaje en pantalla.

```

function OnMouseDown()
{
    if(name == "InventarioInti")
    {
        if(iglesia)
        {
            inventarioTexto.GetComponent(MensajesAyudaJuegoIglesiaScript).MensajeIntiIglesia();
        }

        if(cueva)
        {
            inventarioTexto.GetComponent(MensajesAyudaJuegoIglesiaScript).MensajeIntiCueva();
        }

        if(terreno)
        {
            inventarioTexto.GetComponent(MensajesAyudaJuegoIglesiaScript).MensajeIntiTerreno();
        }
        if(templo)
        {
            cont1++;
            Debug.Log("Entro el valor de cont1: " + cont1);
            if(cont1 == 1)
            {
                imgInti = true;
                inventarioTexto.GetComponent(MensajesAyudaJuegoIglesiaScript).MensajeIntiTemplo(cont1);
            }
            else
            {
                imgInti = false;
                inventarioTexto.GetComponent(MensajesAyudaJuegoIglesiaScript).MensajeIntiTemplo(cont1);
                cont1 = 0;
            }
        }
    }
}

```

#### 4.5.3.7 *Script MensajesAyudaJuegoIglesiaScript.js.*

Este script contiene todas las funciones que manejan los mensajes y que aparecen en la pantalla del jugador a lo largo del juego, todos dependiendo de la escena y el mini juego a resolver.

```

function ActivarImagenAyuda()
{
    guiText.enabled = true;
    guiText.text = "Los INCAS utilizaban\ncalendarios solares\nmuy precisos.\nSelecciona los pilares\nde manera que formen\nun ciclo solar.\nPISTA: Recuerda la Luna es \nel complemento del
}

function ActivarImagenTemplo()
{
    guiText.enabled = true;
    guiText.text = "Busca el IDOLO DEL DIOS INTI.";
}

function ActivarImagenTerreno()
{
    guiText.enabled = true;
    guiText.text = "BUSCA EL TEMPLO I N C A.";
}

function AyudaCuevaMecanismo(numMecanismo : String)
{
    if(numMecanismo == "piedraJuego1")
    {
        guiText.enabled = true;
        guiText.text = "I N T I envio a su hijo \nAYAR MANCO y a su esposa \nMAMA OCLLO para que \ncivilizaran a los humanos.";
    }
    else
    {
        if(numMecanismo == "piedraJuego2")
        {
            {
                guiText.enabled = true;
                guiText.text = "I N T I el DIOS SOL\ncree el cielo y\nla tierra y todo lo que\nen ella existia.";
            }
            else
            {
                if(numMecanismo == "piedraJuego3")
                {
                    {
                        guiText.enabled = true;
                        guiText.text = "En el mismo lugar donde\nse entero el cetro\nfundaron la ciudad de\nCUZCO y asi dio\ninicio al gran imperio INCA.";
                    }
                    else
                    {
                        if(numMecanismo == "piedraJuego4")
                        {
                            {
                                guiText.enabled = true;
                                guiText.text = "Gracias al cetro que\nle otorgo su padre,\nAYAR MANCO y MAMA OCLLO\nsupieron exactamente en donde\ndeberia dar inicio su imperio\ny asi el cetro se ente
                            }
                        }
                    }
                }
            }
        }
    }
}

```

#### 4.5.3.8

#### *Script OpenSceneButtonScript.js.*

Este script funciona únicamente en la pantalla principal del juego, la primera en ver el jugador, en donde se muestra elementos de la historia y el objetivo del juego. La pantalla de título.

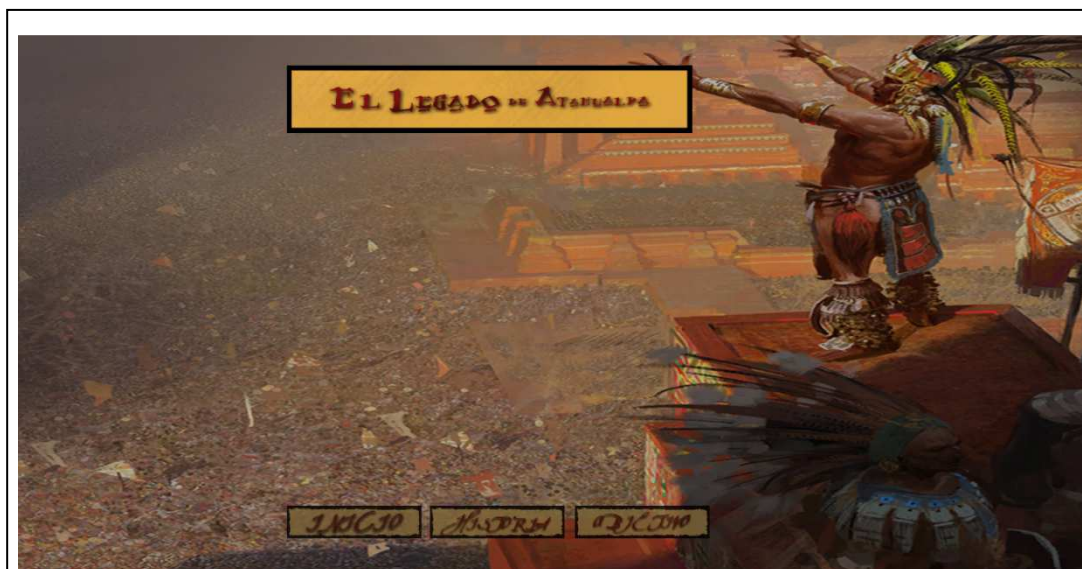


Figura 59. Pantalla del Título del juego.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

En este script se declaran las variables que permitirán observar los paneles que tenga la pantalla y que serán visibles al dar un clic en el botón de “Historia” u “Objetivo”. Para ello estas variables son inicializadas en la función “Awake()”.

Además, se declaran los métodos “OnMouseEnter()” y “OnMouseExit()” que permiten realzar los colores de las imágenes para dar una mayor compresión al jugador de cual botón está a punto de presionar con el cursor del mouse.

```

private var panelMision : GameObject;
private var panelHistoria : GameObject;
private var imagenDeCargado : GameObject;

//funcion Awake funciona al cargarse todos los gameobjects de la escena

function Awake ()
{
    panelMision = GameObject.Find("PanelMision");
    panelHistoria = GameObject.Find("PanelHistoria");
    imagenDeCargado = GameObject.Find("ImagenCargando");
}

function OnMouseEnter ()
{
    guiTexture.color = Color(0.40,0.32,0.25);
}

function OnMouseExit ()
{
    guiTexture.color = Color(0.2, 0.2, 0.2);
}

```

En la función “OnMouseDown()” se revisa en que elemento a dado un clic el jugador, para realizar las funciones correspondientes para cada uno de ellos. Los botones “Historia” y “Objetivo” muestran en pantalla un papel describiendo cada punto relacionado a sus botones.

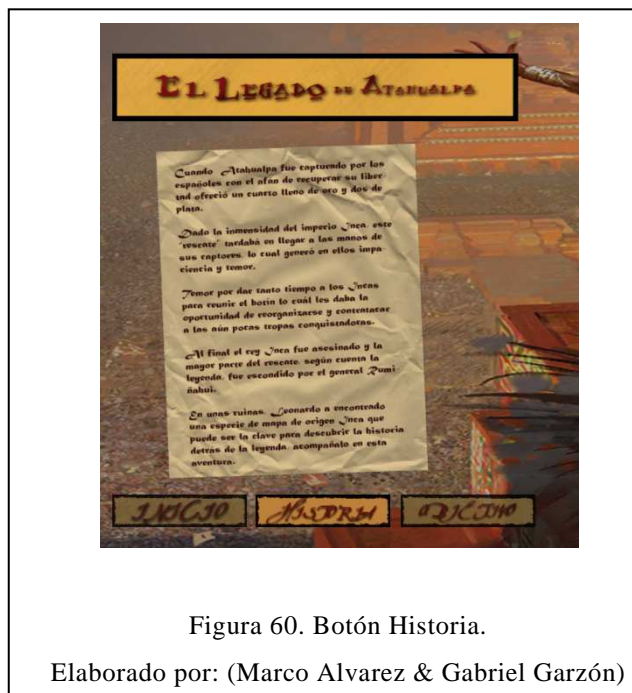
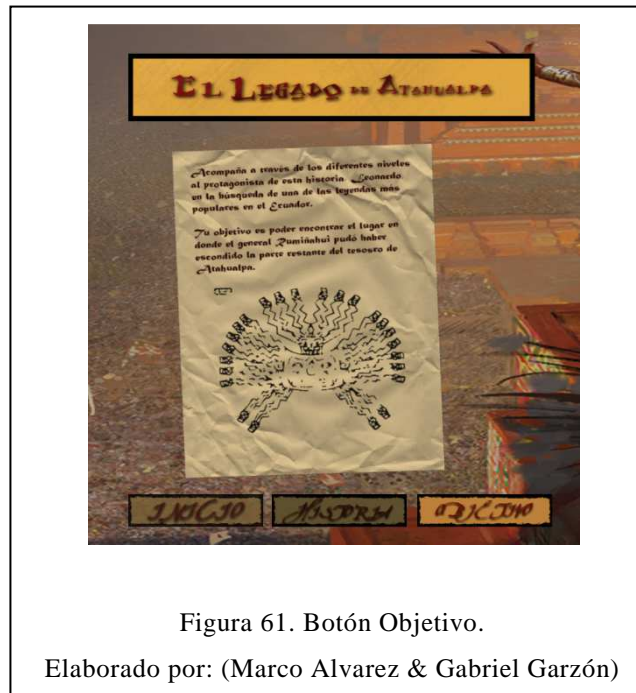


Figura 60. Botón Historia.

Elaborado por: (Marco Alvarez & Gabriel Garzón)



El botón de *Inicio* carga la escena siguiente a la pantalla del título, cuyo nombre es *SceneVideoInicio*.

```
function OnMouseDown()
{
    if(name == "BotonHistoria")
    {
        panelHistoria.guiTexture.enabled = true;
        panelMision.guiTexture.enabled = false;
    }

    if(name == "BotonMision")
    {
        panelMision.guiTexture.enabled = true;
        panelHistoria.guiTexture.enabled = false;
    }

    if(name == "BotonInicio")
    {
        panelHistoria.guiTexture.enabled = false;
        panelHistoria.guiTexture.enabled = false;
        imagenDeCargado.guiTexture.enabled = true;
        Application.LoadLevel("SceneVideoInicio");
    }
}
```

#### 4.5.3.9 Script *PantallaFinalScript.js*.

Este script funciona en la escena final del juego, en ella se puede salir del mismo o volver a empezar.



```

|
function Start()
{
    Screen.lockCursor = false;
}

function OnMouseEnter()
{
    guiTexture.color = Color(0.40,0.32,0.25);
}

function OnMouseExit()
{
    guiTexture.color = Color(0.2, 0.2, 0.2);
}

function OnMouseDown()
{
    if(name == "BotonSalir")
    {
        Application.Quit();
    }

    if(name == "BotonVolverJugar")
    {
        Application.LoadLevel("Scene-Opening");
    }
}
}

```



Figura 62. Pantalla Final del Juego.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

#### 4.5.3.10 Script SantoDomingoActivarMecanismo.js.

Este script inicializa el juego de los pilares, es decir, realiza la animación inicial del juego en donde los pilares de la iglesia se elevan revelando las imágenes en sus paredes.

```
//En este script se pretende generar la animación del mecanismo

var activadorJuego : GameObject;
var audioActivadorJuego : GameObject;
var pilar1 : GameObject;
var pilar2 : GameObject;
var pilar3 : GameObject;
var pilar4 : GameObject;

function OnTriggerEnter (other:Collider)
{
    iTween.MoveTo(activadorJuego, Vector3(0, -0.01, 0), 7);
    audioActivadorJuego.audio.Play();
    ElevarPilaresIglesia();
}

function ElevarPilaresIglesia()
{
    iTween.MoveTo(pilar1, Vector3(0,5.38,0), 7);
    iTween.MoveTo(pilar2, Vector3(0,5.38,0), 7);
    iTween.MoveTo(pilar3, Vector3(0,5.38,0), 7);
    iTween.MoveTo(pilar4, Vector3(0,5.38,0), 7);
}
}
```

#### 4.5.3.11 *Script SantoDomingoControlTriggers.js.*

Este script controla los diferentes “Triggers” que se encuentran a lo largo de las escenas del juego.

Un trigger es un elemento dentro de una escena, es invisible para el jugador y se lo puede representar de la siguiente manera en Unity.

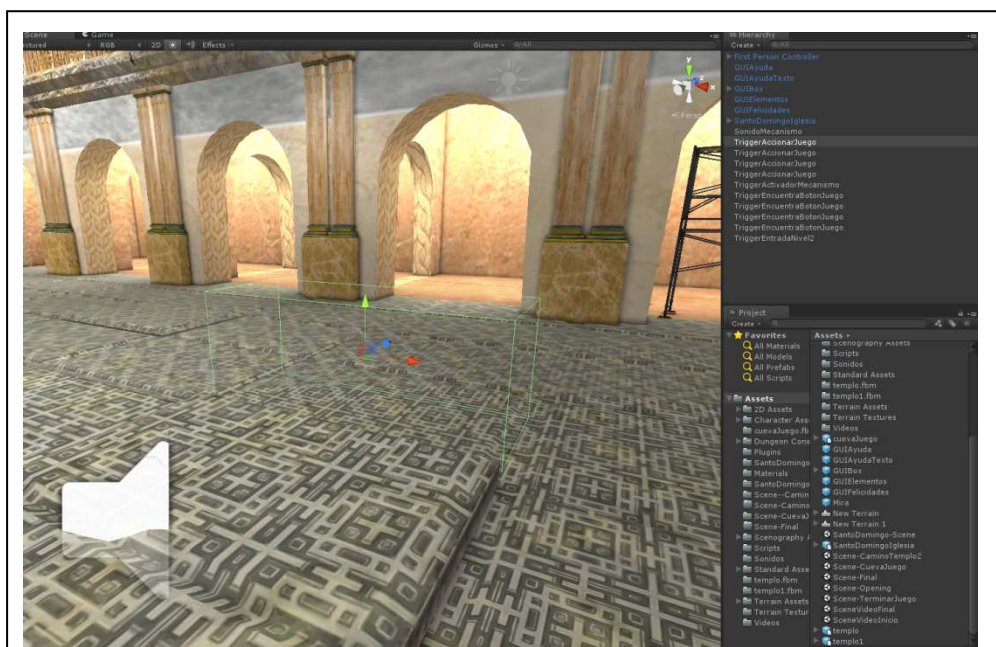


Figura 63. Trigger en Unity

Elaborado por: (Marco Alvarez & Gabriel Garzón)

El uso de estos componentes es muy variado, en el juego se los utiliza para mostrar las ayudas generales en el juego, el jugador al toparse con uno de estos elementos mostrará el mensaje relacionado a este elemento y que está declarado en el script.

Para nuestro ejemplo se tiene al elemento “TriggerAccionarJuego”, en el script hay una función relacionada al nombre de este elemento, como se puede ver en la siguiente imagen.

```
function Awake()
{
    guiTextObject = GameObject.Find("GUIElementos");
}

function OnTriggerEnter(other:Collider)
{
    if (name == "TriggerEncuentraBotonJuego")
    {
        guiTextObject.GetComponent(SantoDomingoUITextScript).AccionarJuegoIglesia();
    }

    if (name == "TriggerAccionarJuego")
    {
        guiTextObject.GetComponent(SantoDomingoUITextScript).MensajeIniciarJuego();
    }

    if (name == "TriggerReiniciarJuego")
    {
        guiTextObject.GetComponent(SantoDomingoUITextScript).MensajeReiniciarJuegoIglesia();
    }

    if (name == "TriggerEntradaNivel2")
    {
        Application.LoadLevel("Scene-CuevaJuego");
    }

    if (name == "TriggerTextos")
    {
        guiTextObject.GetComponent(SantoDomingoUITextScript).MensajeBuscarJuegoCueva();
    }

    if (name == "TriggerActivadorJuegoCueva")
    {
        guiTextObject.GetComponent(SantoDomingoUITextScript).PistaJuegoCueva();
    }

    if (name == "TriggerInicioNivel3")
    {
        guiTextObject.GetComponent(SantoDomingoUITextScript).MensajeJuegoTerreno();
    }

    if (name == "TriggerAccesoNivelFinal")
    {
        Application.LoadLevel("Scene-Final");
    }
}
```

Esto funciona cuando el jugador tiene contacto con alguno de estos elementos, se utiliza la función “OnTriggerEnter(other:Collider)” la cual siempre comienza a trabajar al ocurrir dicho evento de colisión o contacto. En este ejemplo a la variable “guiTextObject”, previamente inicializada, muestra el mensaje contenido en el script “SantoDomingoUITextScript” en el método “MensajeIniciarJuego()”.

#### 4.5.3.12 *Script SantoDomingoGUITextScript.js.*

Este script contiene los métodos que son los que muestran los diferentes textos en pantalla para ayuda general del jugador, estos métodos tienen una peculiaridad, los mensajes que muestran tienen un tiempo máximo para ser visualizados, es decir, pasado el tiempo estipulado en el código los mensajes desaparecerán hasta que nuevamente sean llamados. Como se ha visto en diferentes partes de los otros scripts.

---

```
//Este script controlará los mensajes que pueden salir en la pantalla del jugador

function AccionarJuegoIglesia()
{
    guiText.enabled = true;
    guiText.text = "Encuentra el mecanismo que acciona el secreto de la iglesia.";
    yield WaitForSeconds(10);
    guiText.enabled = false;
}

function MensajeBuscarJuegoCueva()
{
    guiText.enabled = true;
    guiText.text = "Haz ingresado a la parte inferior de la iglesia \n Busca el siguiente desafio";
    yield WaitForSeconds(5);
    guiText.enabled = false;
}

function MensajeCaminoSinSalida()
{
    guiText.enabled = true;
    guiText.text = "Este camino no tiene salida";
    yield WaitForSeconds(10);
    guiText.enabled = false;
}

function MensajeEscenaFinal()
{
    guiText.enabled = true;
    guiText.text = "Encuentra la representacion del Dios INCA y descubre \n EL LEGADO DE ATAHUALPA";
    yield WaitForSeconds(10);
    guiText.enabled = false;
}

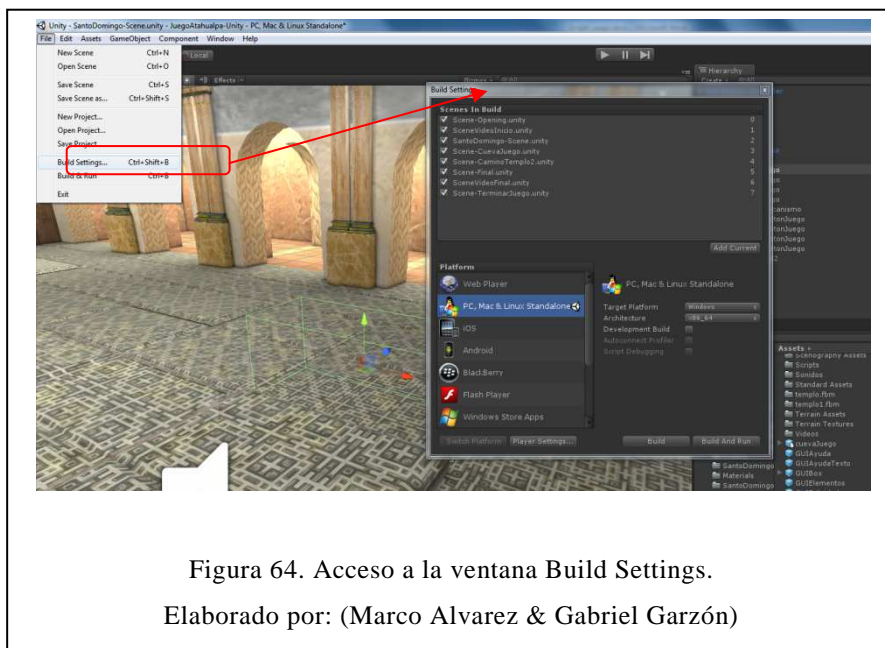
function MensajeIniciarJuego()
{
    guiText.enabled = true;
    guiText.text = "Colocate encima del mecanismo del piso para iniciar el juego";
    yield WaitForSeconds(10);
    guiText.enabled = false;
}

function MensajeJuegoResuelto()
{
    guiText.enabled = true;
    guiText.text = "Haz logrado la secuencia correcta!!";
    yield WaitForSeconds(10);
    guiText.enabled = false;
}
```

#### 4.5.4 Construcción del videojuego.

Para poder tener un flujo entre las diferentes escenas del juego, se debe especificar en Unity el orden en el cual serán llamadas.

Para ello se debe acceder a la ventana “Build Settings”.



En esta pantalla se puede seleccionar el orden en el que las escenas que se han creado en el proyecto serán procesadas. Como se puede observar también se puede elegir la plataforma final del juego, esta puede ser las consolas de sobremesa o computadores como PC, Mac o incluso Linux.

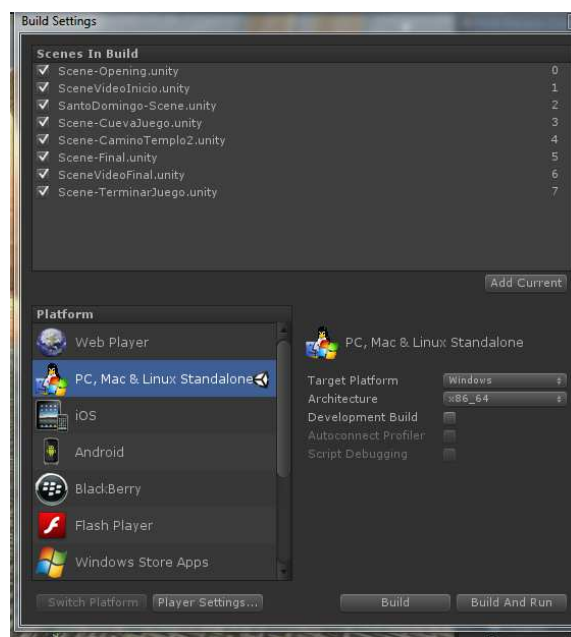


Figura 65. Orden de las escenas del juego y plataforma final para el mismo.  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

Una vez realizado esto se puede probar el juego antes de crear el producto final. Para ello se debe regresar a la interfaz de Unity y se da un clic en el botón “Play” en la parte superior de la pantalla.

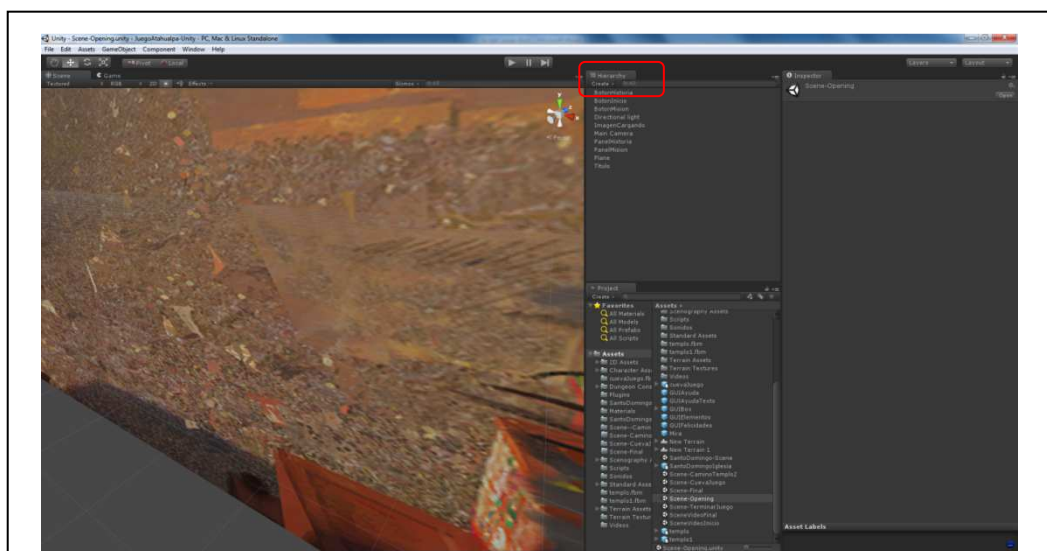


Figura 66. Botón play en la interfaz de Unity.  
Elaborado por: (Marco Alvarez & Gabriel Garzón)



Una vez hecho esto se iniciará el juego en la escena en la que el desarrollador se encuentre, en este caso será la primera, es decir la que muestra en pantalla el título del juego.

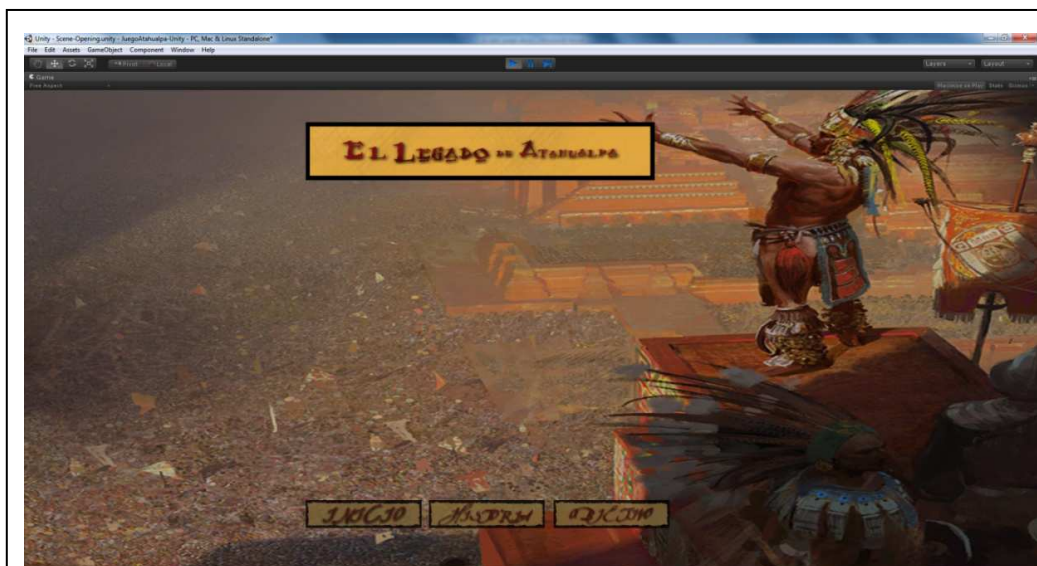


Figura 67. Pantalla de inicio del juego.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Los botones de Inicio, Historia y Objetivos pueden ser presionados. El primero iniciara el juego con un video introductorio del mismo, el segundo mostrará en pantalla el documento que contiene la historia que se desarrolla el juego y el tercero mostrará en pantalla el documento que explica el objetivo final del juego.

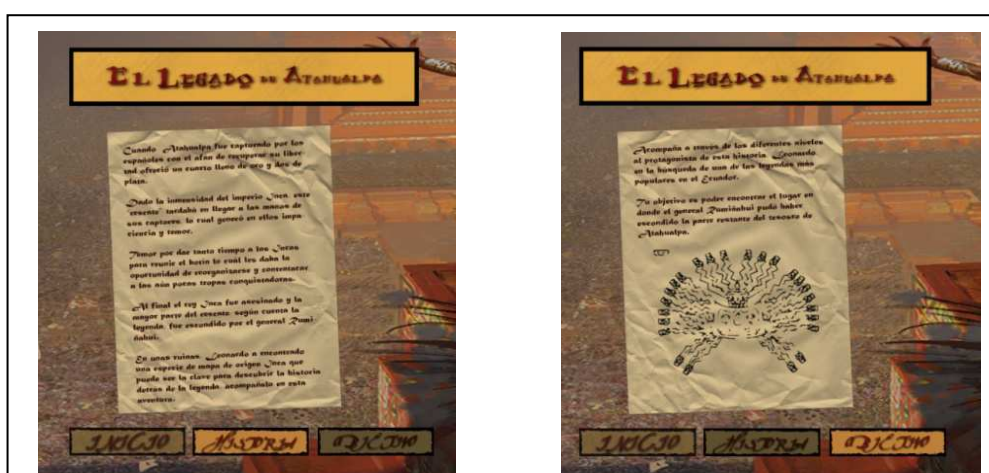
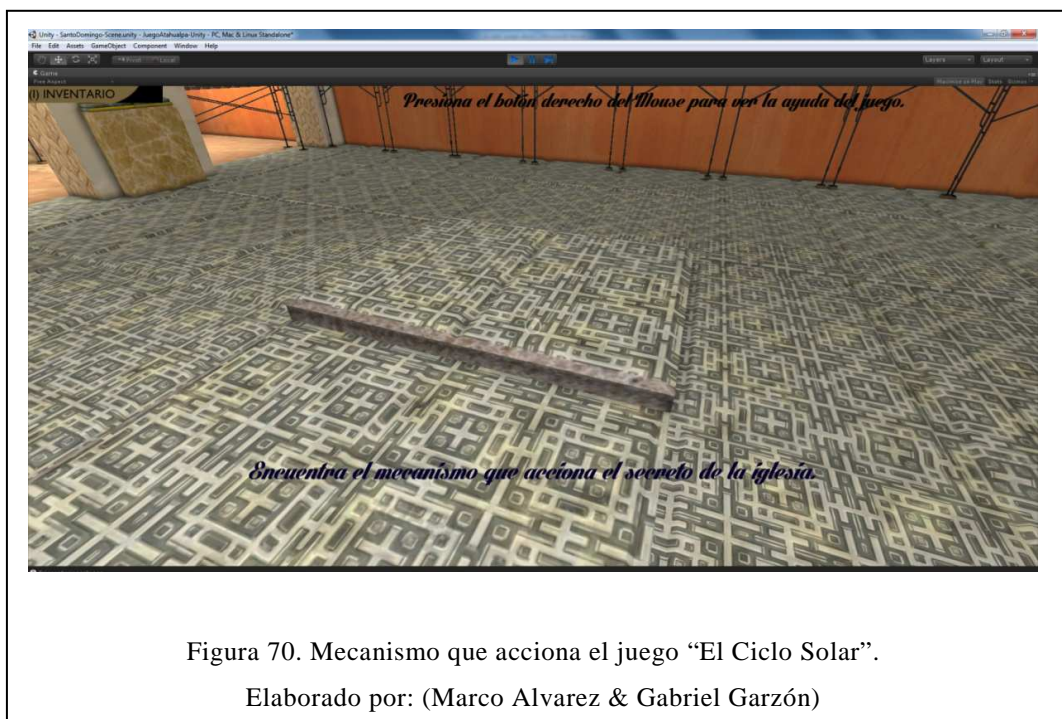


Figura 68. Botones de pantalla de título.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

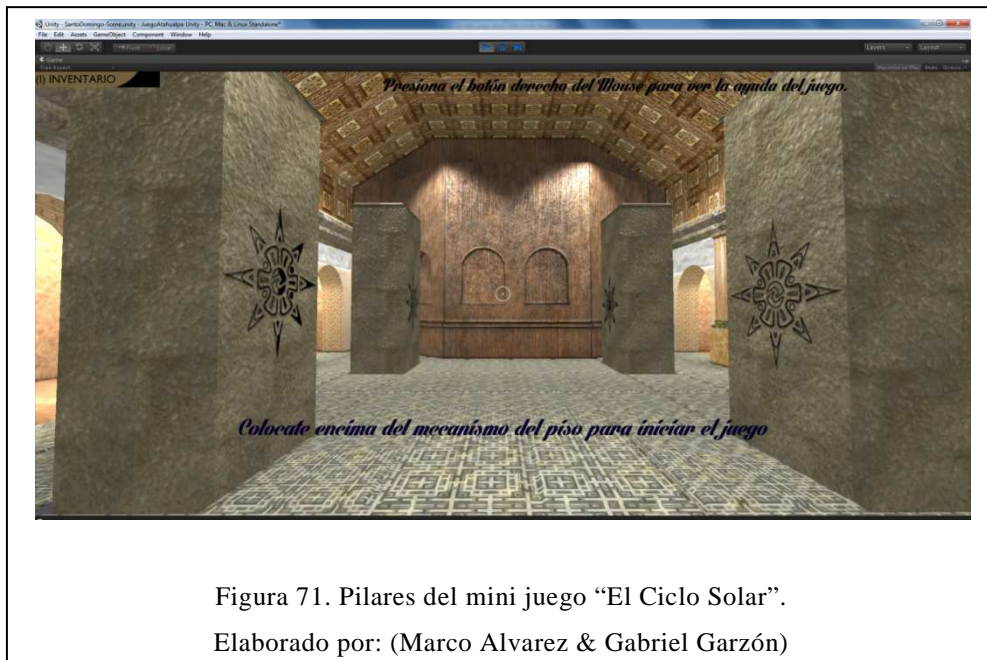


Después del video introductorio, el jugador es trasladado a la iglesia de Santo Domingo en donde podrá encontrar el primer videojuego. “El Ciclo Solar”, en esta escena se tiene que encontrar el mecanismo que accione el mini juego, el cual está colocado en el piso y es fácil de notar.





El jugador deberá situarse encima de este mecanismo, esto activará el mini juego, haciendo que los pilares que se encontraban ocultos aparezcan en escena.



Ahora el jugador podrá seleccionar un pilar, solamente tendrá que poner la mirilla que se encuentra en la pantalla y apuntar al pilar que quiera elegir, si se da un clic sobre un pilar este regresará a su posición inicial.



Para poder resolver este mini juego el jugador necesitará elegir los pilares en secuencia de un ciclo solar, para dar una pista al jugador se habilita la ayuda que puede ser vista al dar un clic derecho en la pantalla del juego.

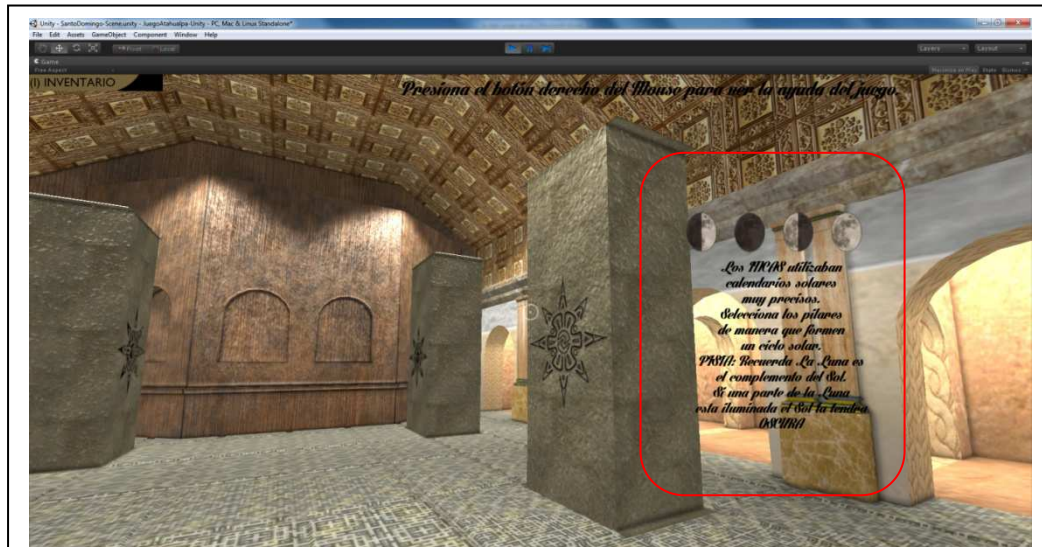


Figura 73. Ayuda para el primer mini juego.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Si el jugador escoge correctamente los pilares, se abrirá una puerta en la mitad del piso, que será la puerta hacia la nueva escena, caso contrario el jugador tendrá que volver a accionar el juego para volver a elegir los pilares.

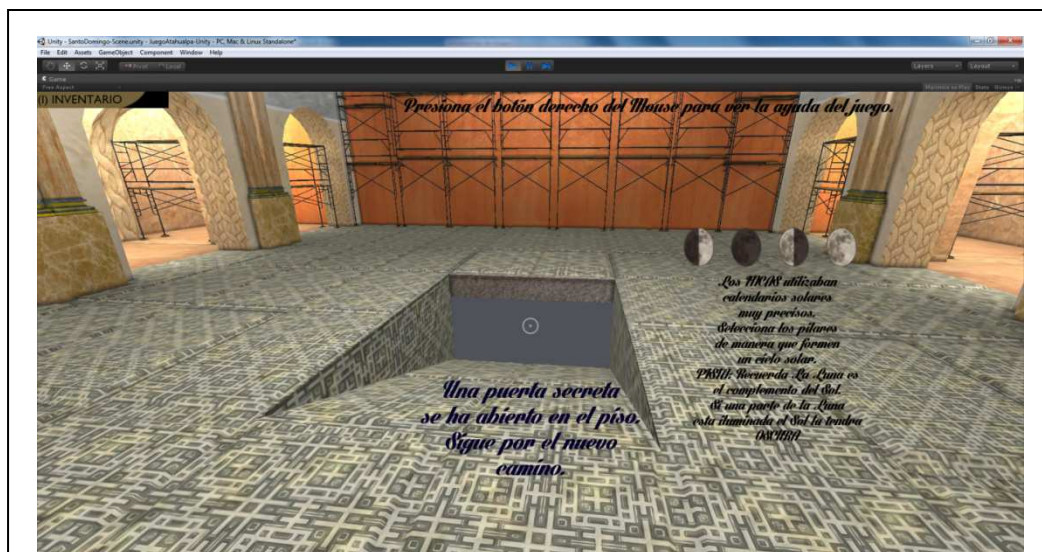


Figura 74. Mini juego “El Ciclo Solar” ha sido resuelto.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Al ingresar por este acceso se cargará la siguiente escena en la cual se debe resolver otro mini juego “La Historia del Primer Inca”.

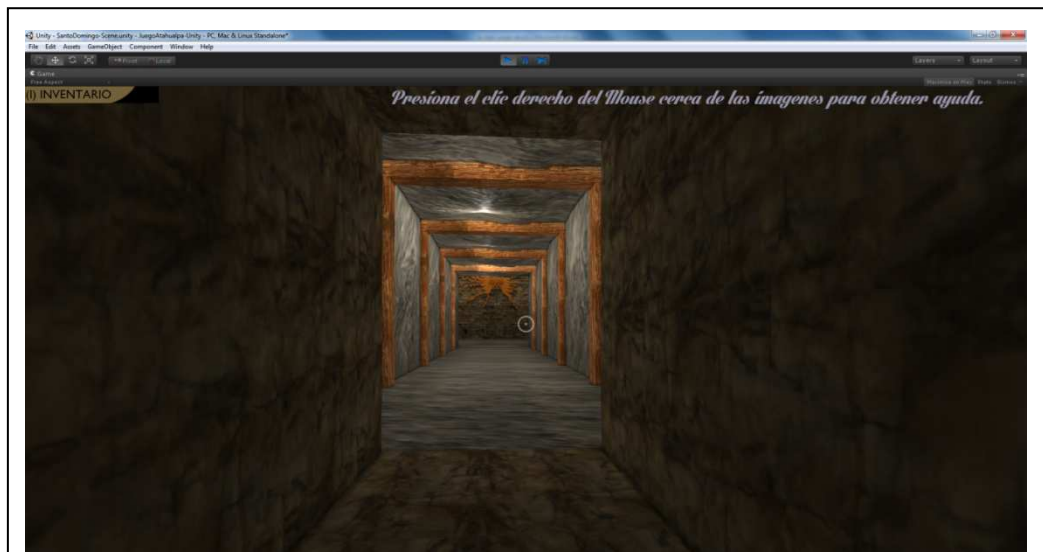


Figura 75. Inicio de escena.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

En este mini juego se debe elegir el orden correcto de las piedras incrustadas en la pared y así contar la historia del primer inca Ayar Manco.



Figura 76. Elementos del mini juego “La historia del primer Inca”.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

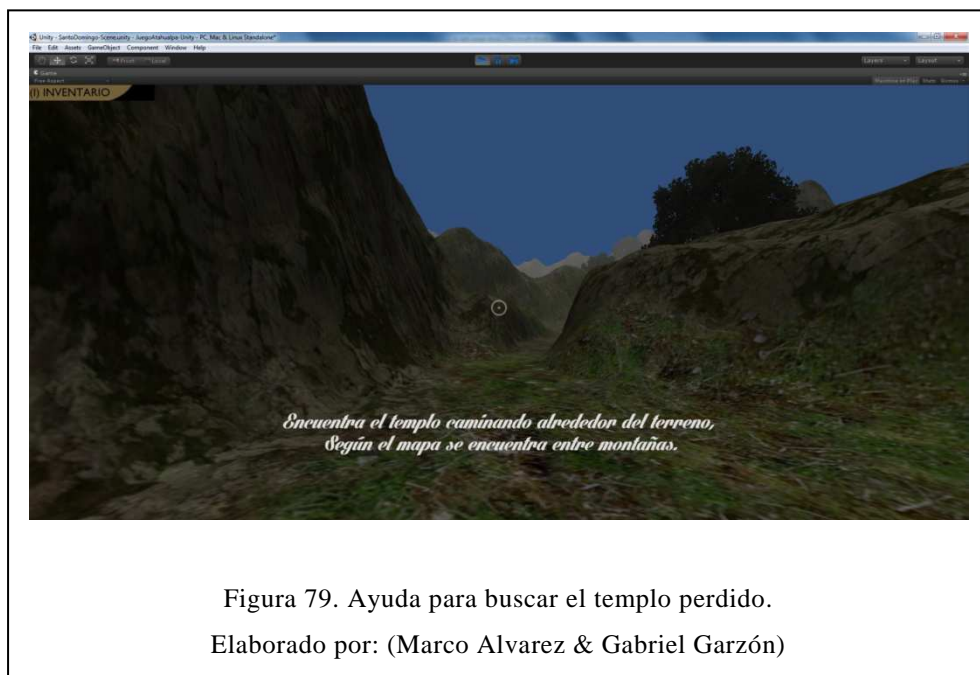


Figura 77. Diferentes ayudas relacionadas a las imágenes en frente del jugador.

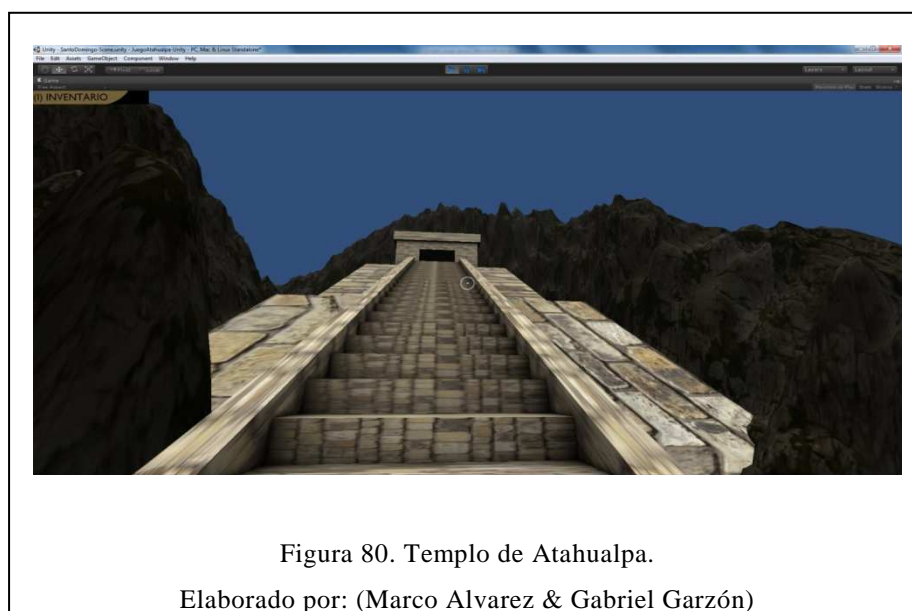
Elaborado por: (Marco Alvarez & Gabriel Garzón)

Una vez tomada la imagen del Dios Inti, el jugador será trasladado a un escenario amplio, representa un bosque escondido en el cual el objetivo principal del jugador es encontrar el templo perdido de Atahualpa.

A lo largo del camino se recibirá ayuda por parte de mensajes que aparecerán en la pantalla del jugador.



Al encontrar el templo se debe subir por las escaleras del mismo para acceder a su interior.



Al ingresar al templo el jugador se encontrará en una estancia a oscuras, podrá ver gracias a un haz de luz que representa la linterna que se tiene en el inventario. El objetivo en esta escena es encontrar la representación del Dios Inti y mostrar el objeto que se obtuvo en un nivel anterior.

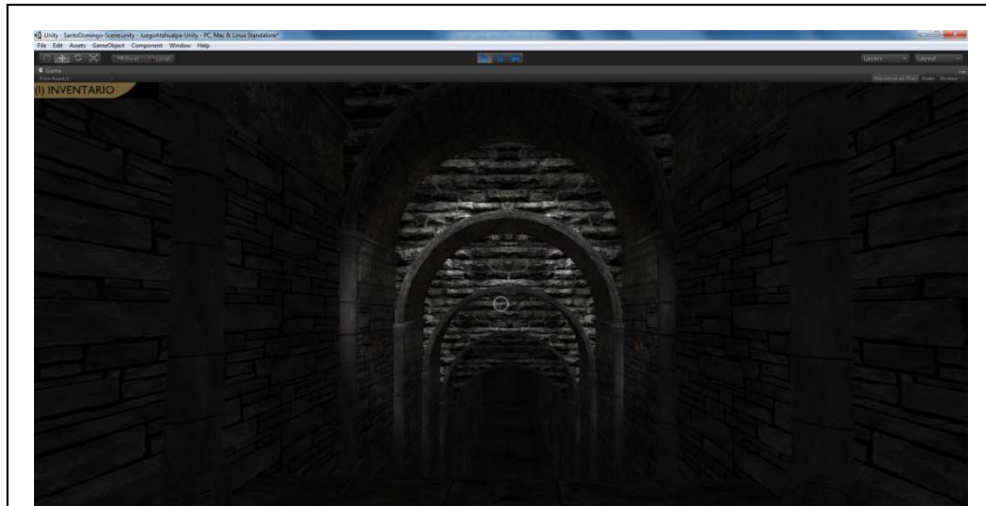


Figura 81. Interior Templo.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Al encontrar la representación, se debe mostrar la que el jugador tiene, para ello tiene que ingresar al inventario presionando la tecla “I” y elegir la imagen del Dios Inca.

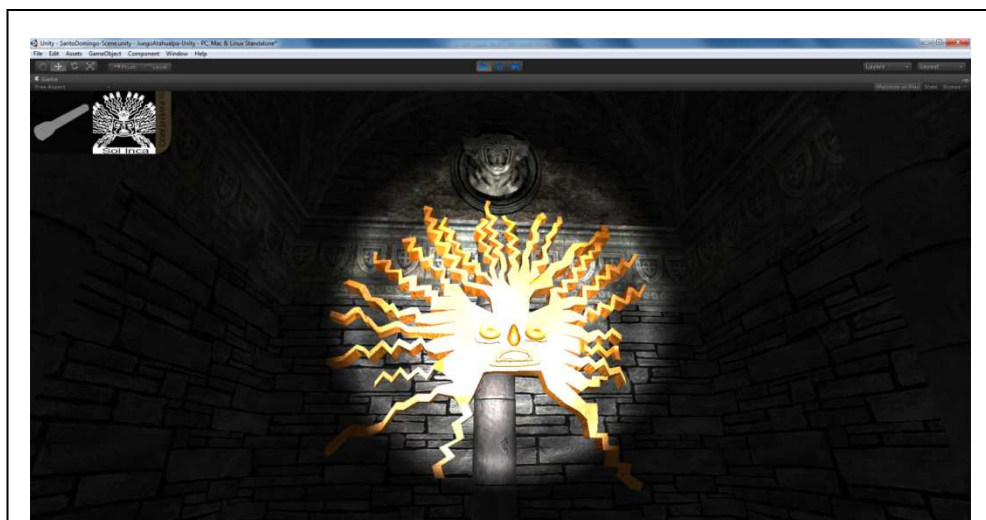


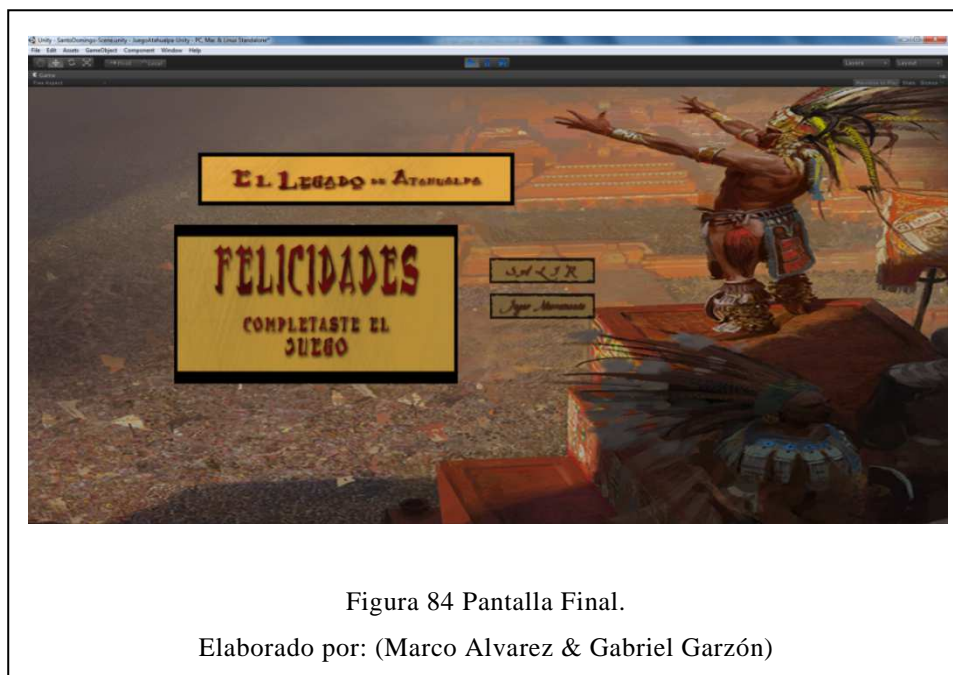
Figura 82. Selección del Dios Inca.

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Al seleccionar la imagen se mostrará el video final del juego, en el cual se encuentra el verdadero mensaje del juego. Lo que buscaba el jugador puede ser oro y riquezas, pero lo que encuentra es sabiduría de parte de los Incas, el video muestra como una traición puede aniquilar el imperio más grande conocido en América.



Finalmente se mostrará en pantalla la escena final en donde se podrá elegir en volver a repetir el juego o por el contrario salir de él.



#### 4.5.4.1 *Script FuncionalidadLibro.cs.*

Este script permite la interacción del jugador con la herramienta libro. Esta herramienta simulará un libro de anotaciones en donde se podrá encontrar un resumen de los mini juegos resueltos así como también de la historia del juego.

Este script se diferencia de los demás porque no es un archivo con lenguaje base Java script, este script se lo realizo en el lenguaje C# con el fin de demostrar que se puede realizar un proyecto en Unity con los dos lenguajes. Pero siempre hay que tener en cuenta que un archivo escrito en C# no podrá comunicarse con un archivo escrito en Java script. Es decir si se necesita algún valor de una variable de un archivo C# será imposible tenerla como referencia en un archivo Javacript y viceversa.

Este script comienza con la llamada de las librerías que se necesitan para la funcionalidad del mismo.

```
using UnityEngine;  
using System.Collections;  
using System.Collections.Generic;
```

Luego se declara la clase que será el mismo nombre del archivo, es decir FuncionalidadLibro.

```
public class FuncionalidadLibro : MonoBehaviour
```

Luego se tiene la declaración de las variables a utilizar en el script, todas ellas son variables públicas, por lo cual se podrá manipular sus valores desde el panel de control de Unity directamente, esto permitirá cambiar las imágenes dependiendo de la escena en la que el jugador se encuentre.

La variable tipo lista “paginasLibro” se controla igual que si fuera un array, es decir, para poder obtener un valor de esta variable se debe determinar la posición del elemento de la lista.



```
//variables
public List<Texture2D> paginasLibro = new List<Texture2D>();
public Texture2D flechaIzq;
public Texture2D flechaDer;
public bool mostrarLibreta;
public int contClicks;
public int currentImage;
public AudioClip sonidoLibro;
```

En la función “Start” inicializará algunas de las variables, “mostrarLibreta” es una variable booleana la cual, dependiendo de su valor, mostrará las imágenes de la libreta. La variable “contClicks” se refiere a la que controlará que al hacer clic sobre la herramienta cambie el valor de la variable “mostrarLibreta”, por último “currentImage” obtendrá el valor de la posición en la lista “paginasLibro” de la imagen que en ese momento estará viendo el jugador.

```
// Use this for initialization
void Start ()
{
    mostrarLibreta = false;
    contClicks = 0;
    currentImage = 0;
}
```

La función “Update” controlará que al dejar de utilizar la herramienta, las imágenes desaparezcan de la pantalla.

```
void Update()
{
    if(Screen.lockCursor)
    {
        contClicks = 0;
        mostrarLibreta = false;
    }
}
```

La función “OnMouseDown” controla los clics que realiza el jugador sobre la herramienta, al hacer clic añade una unidad al valor de la variable “contClicks”, siempre que esta variable tenga el valor de “1” se permitirá mostrar las imágenes de la libreta, dado que la variable “mostrarLibreta” tendrá valor de “true”, con cualquier otro valor de “contClicks” esta variable tendrá el valor de “false”.

```

void OnMouseDown()
{
    //esta variable servira como parametro para mostrar la libreta o no
    contClicks++;
    if(contClicks == 1)
    {
        mostrarLibreta = true;
        currentImage = 0;
    }
    else
    {
        mostrarLibreta = false;
        contClicks = 0;
    }
}

```

La siguiente función “OnGUI” es la que “dibuja” las imágenes que tenga el libro así como también las imágenes de las variables “flechaIzq” y “flechaDer” que permitirá el transcurso de las imágenes del libro.

Si la variable “mostrarLibreta” tiene un valor de “true” se declara condiciones que revisan si se hace un clic sobre las imágenes de flechas, si se da un clic a “flechaIzq” se resta una unidad a la variable “currentImage” pero se sumará una unidad si se da un clic sobre “flechaDer”.

Las siguientes líneas controlan que el valor en “currentImage” no sobrepase la capacidad de la lista. Por último con la función “GUI.DrawTexture” dibuja las imágenes del libro.

```

//metodo para Crear interfaces
void OnGUI()
{
    if(mostrarLibreta)
    {
        if(GUI.Button(new Rect(Screen.width * 0.15f, Screen.height * 0.5f, flechaIzq.width, flechaIzq.height), flechaIzq))
        {
            currentImage--;
            if(currentImage < 0)
            {
                currentImage = 0;
            }
            else
            {
                audio.clip = sonidoLibro;
                audio.Play();
            }
        }
        if(GUI.Button(new Rect(Screen.width * 0.8f, Screen.height * 0.5f, flechaDer.width, flechaDer.height), flechaDer))
        {
            currentImage++;
            if(currentImage >= paginasLibro.Capacity)
            {
                currentImage = paginasLibro.Capacity - 1;
            }
            else
            {
                audio.clip = sonidoLibro;
                audio.Play();
            }
        }
        GUI.DrawTexture(new Rect(Screen.width * 0.25f, Screen.height * 0.2f, paginasLibro[currentImage].width, paginasLibro[currentImage].height), paginasLibro[currentImage]);
    }
}

```

La forma en cómo se comporta este script es la siguiente, en Unity se asigna el script al GameObject “InventarioLibro”.

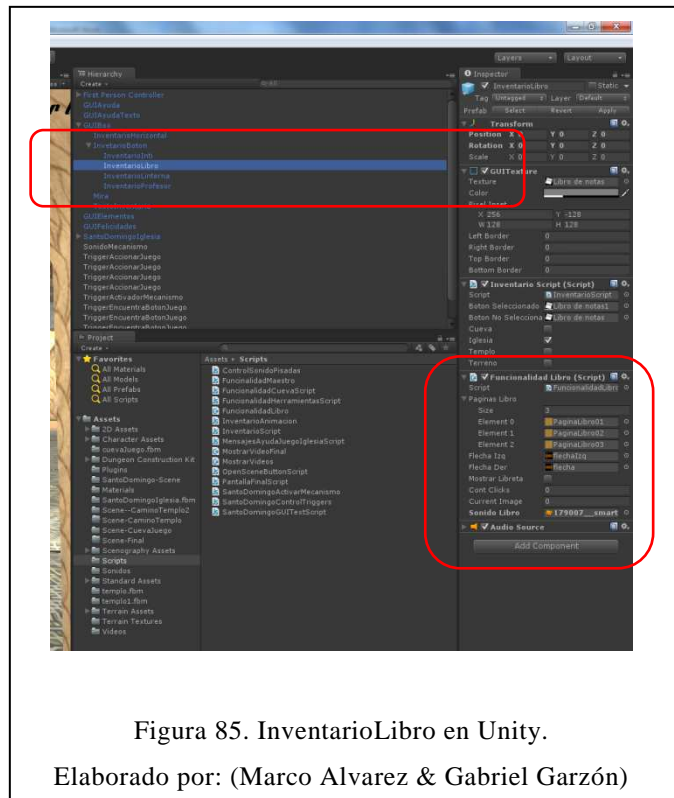
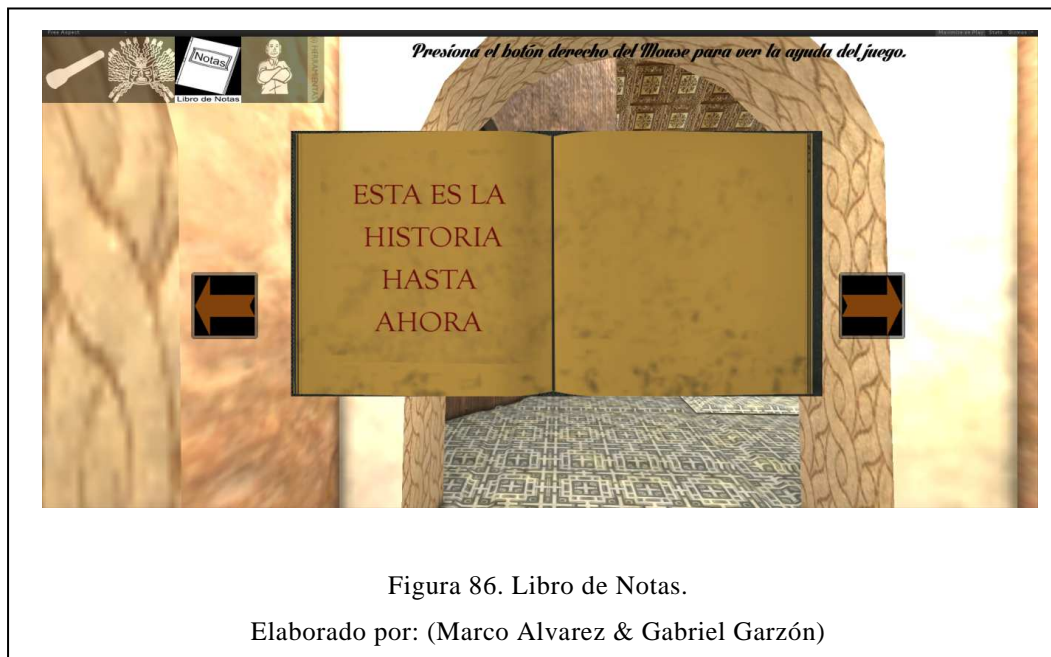


Figura 85. InventarioLibro en Unity.  
Elaborado por: (Marco Alvarez & Gabriel Garzón)

En Unity se puede cambiar los valores de las variables públicas, en el caso de la lista se debe ingresar cuantas imágenes se quiere controlar, es decir, la capacidad o “Size” de la lista, una vez ingresado este valor se debe de arrastrar las imágenes que se encuentran en nuestro proyecto a los casilleros con el nombre “Element”. Lo mismo se realizará con las imágenes de las flechas y con el archivo de audio “Sonido Libro”.

Una vez en el juego, se debe ingresar a las herramientas y dar un clic sobre el elemento “Libro de Notas” esto permitirá ver las imágenes que se han asignado a la lista y se podrá cambiar de imagen si se da un clic sobre las flechas.



#### 4.5.4.2 Script *FuncionalidadMaestro.js*.

En el campo de herramientas se encuentra el botón de “Profesor” el cual debe simular la comunicación de un maestro con el jugador, de esta manera tendrá una ayuda extra para la resolución de problemas.

Este script cumple las mismas funciones que el script descrito anteriormente, solo que este archivo tiene como lenguaje Java script dado que se necesita conocer valores externos a este archivo y que se encuentran en otros scripts.

Primero se importa la librería necesaria para controlar listas de elementos y a continuación se crean las variables a utilizar.

```

1  #pragma strict
2  import System.Collections.Generic;
3
4  //variables
5
6  public var listaImagenes : List.<Texture2D>;
7  public var listaImagenes2 : List.<Texture2D>;
8  public var flechaIzq : Texture2D;
9  public var flechaDer : Texture2D;
10 public var esTemplo : boolean;
11 public var esCueva : boolean;
12 public var esTerreno : boolean;
13 public var esInteriorTemplo : boolean;
14 private var mostrarAyudaTemplo1 : boolean;
15 private var mostrarAyudaTemplo2 : boolean;
16 private var mostrarAyudaCueva1 : boolean;
17 private var mostrarAyudaCueva2 : boolean;
18 private var mostrarAyudaTerreno1 : boolean;
19 private var mostrarAyudaTerreno2 : boolean;
20 private var mostrarAyudaTemploP1 : boolean;
21 private var mostrarAyudaTemploP2 : boolean;
22 public var objActivador : SantoDomingoActivarMecanismo;
23 public var triggerActivador : GameObject;
24 private var currentImage : int;
25 private var cClicks : int;

```

La función “Start” inicializará variables dependiendo del escenario en el que se encuentre el jugador.

```
function Start ()
{
    if(esTemplo)
    {
        triggerActivador = GameObject.Find("TriggerActivadorMecanismo");
        objActivador = triggerActivador.GetComponent("SantoDomingoActivarMecanismo");
        Debug.Log("El objetivo activador se creo: " + objActivador.estaActivado);
        cClicks = 0;
    }

    if(esCueva)
    {
        triggerActivador = GameObject.Find("caraIntiCueva");
    }

    if(esTerreno)
    {
        triggerActivador = GameObject.Find("templo1");
    }

    if(esInteriorTemplo)
    {
        triggerActivador = GameObject.Find("caraInti");
    }
}
```

La función “Update” controla, dependiendo del escenario en que se encuentre el jugador, mostrar las imágenes referentes al profesor.

```
function Update()
{
    if(Screen.lockCursor)
    {
        if(esTemplo)
        {
            mostrarAyudaTemplo1 = false;
            mostrarAyudaTemplo2 = false;
        }
        if(esCueva)
        {
            mostrarAyudaCueva1 = false;
            mostrarAyudaCueva2 = false;
        }
        if(esTerreno)
        {
            mostrarAyudaTerreno1 = false;
            mostrarAyudaTerreno2 = false;
        }

        if(esInteriorTemplo)
        {
            mostrarAyudaTemploP1 = false;
            mostrarAyudaTemploP2 = false;
        }
        cClicks = 0;
    }
}
```

En este script se utilizan dos listas “listaImágenes” y “listaImágenes2” dado que las imágenes deben cambiar dependiendo de la situación en la que se encuentre el jugador, por ejemplo, si necesita resolver un mini-juego el profesor debe ayudarlo brindándole pistas, pero si el usuario se encuentra comenzando el nivel, el profesor debe animarle para que investigue el escenario en el que se encuentra.

Es por ello que se utiliza, en cada escenario, dos tipos de variables booleanas que determinan que lista de imágenes mostrar al jugador, esto se controla al dar un clic sobre la herramienta “Profesor” a través del método “OnMouseDown”.

```
function OnMouseDown ()
{
    currentImage = 0;
    cClicks++;
    if(cClicks == 1)
    {
        if(esTemplo)
        {
            if(objActivador.estaActivado)
            {
                mostrarAyudaTemplo1 = false;
                mostrarAyudaTemplo2 = true;
            }
            else
            {
                mostrarAyudaTemplo1 = true;
                mostrarAyudaTemplo2 = false;
            }
        }
    }

    //Escena de la cueva
    if(esCueva)
    {
        var personajeJuego : GameObject;
        personajeJuego = GameObject.Find("First Person Controller");

        var distancia : float;
        distancia = Vector3.Distance(triggerActivador.transform.position, personajeJuego.transform.position);

        if(distancia > 10f)
        {
            mostrarAyudaCueva1 = true;
            mostrarAyudaCueva2 = false;
        }
        else
        {
            mostrarAyudaCueva1 = false;
            mostrarAyudaCueva2 = true;
        }
    }
}

//Escena del terreno
if(esTerreno)
{
    personajeJuego = GameObject.Find("First Person Controller");

    distancia = Vector3.Distance(triggerActivador.transform.position, personajeJuego.transform.position);
    Debug.Log("La distancia al templo es: " + distancia);

    if(distancia > 40f)
    {
        mostrarAyudaTerreno1 = true;
        mostrarAyudaTerreno2 = false;
    }
    else
    {
        mostrarAyudaTerreno1 = false;
        mostrarAyudaTerreno2 = true;
    }
}

//Escena del templo
if(esInteriorTemplo)
{
    personajeJuego = GameObject.Find("First Person Controller");

    distancia = Vector3.Distance(triggerActivador.transform.position, personajeJuego.transform.position);
    Debug.Log("La distancia al templo es: " + distancia);

    if(distancia > 10f)
    {
        mostrarAyudaTemploP1 = true;
        mostrarAyudaTemploP2 = false;
    }
    else
    {
        mostrarAyudaTemploP1 = false;
        mostrarAyudaTemploP2 = true;
    }
}
}
```

```

else
{
    if (esTemplo)
    {
        mostrarAyudaTemplo1 = false;
        mostrarAyudaTemplo2 = false;
    }
    /******/
    if (esCueva)
    {
        mostrarAyudaCueva1 = false;
        mostrarAyudaCueva2 = false;
    }
    /******/
    if (esTerreno)
    {
        mostrarAyudaTerreno1 = false;
        mostrarAyudaTerreno2 = false;
    }
    /******/
    if (esInteriorTemplo)
    {
        mostrarAyudaTemploP1 = false;
        mostrarAyudaTemploP2 = false;
    }
    cClicks = 0;
}

```

Dependiendo del escenario en el que se encuentre el jugador los valores de las variables booleanas cambiarán, por ejemplo, en el escenario de la iglesia de Santo Domingo una vez que el mini-juego haya sido activado las imágenes del profesor deberán cambiar, pero en los demás escenarios dependerá de la distancia que se encuentre el jugador a un determinado objeto.

La función “OnGUI” cumple el mismo funcionamiento que en el script “FuncionalidadLibro.cs” en la cual dibujará las imágenes en la pantalla.

```

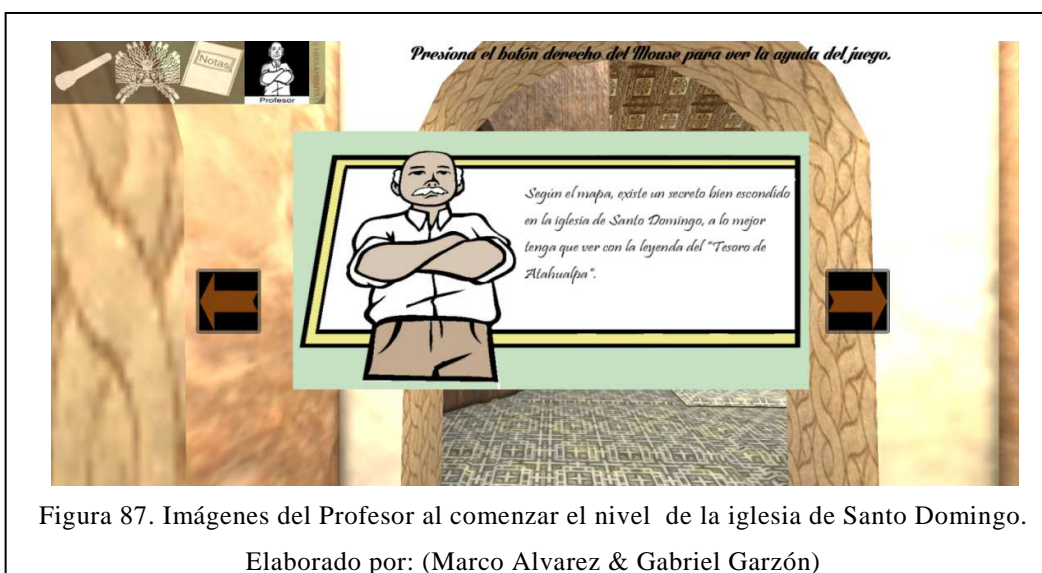
function OnGUI()
{
    if(mostrarAyudaTemplo1 || mostrarAyudaCueva1 || mostrarAyudaTerreno1 || mostrarAyudaTemploP1)
    {
        if(GUI.Button(Rect(Screen.width * 0.15f, Screen.height * 0.5f, flechaIzq.width, flechaIzq.height), flechaIzq))
        {
            currentImage--;
            if(currentImage < 0)
            {
                currentImage = 0;
            }
        }
        if(GUI.Button(Rect(Screen.width * 0.8f, Screen.height * 0.5f, flechaDer.width, flechaDer.height), flechaDer))
        {
            currentImage++;
            if(currentImage >= listaImagenes.Capacity)
            {
                currentImage = listaImagenes.Capacity - 1;
            }
        }
        GUI.DrawTexture(Rect(Screen.width * 0.25f, Screen.height * 0.2f, listaImagenes[currentImage].width, listaImagenes[currentImage].height), listaImagenes[currentImage]);
    }

    if(mostrarAyudaTemplo2 || mostrarAyudaCueva2 || mostrarAyudaTerreno2 || mostrarAyudaTemploP2)
    {
        if(GUI.Button(Rect(Screen.width * 0.15f, Screen.height * 0.5f, flechaIzq.width, flechaIzq.height), flechaIzq))
        {
            currentImage--;
            if(currentImage < 0)
            {
                currentImage = 0;
            }
        }
        if(GUI.Button(Rect(Screen.width * 0.8f, Screen.height * 0.5f, flechaDer.width, flechaDer.height), flechaDer))
        {
            currentImage++;
            if(currentImage >= listaImagenes2.Capacity)
            {
                currentImage = listaImagenes2.Capacity - 1;
            }
        }
        GUI.DrawTexture(Rect(Screen.width * 0.25f, Screen.height * 0.2f, listaImagenes2[currentImage].width, listaImagenes2[currentImage].height), listaImagenes2[currentImage]);
    }
}

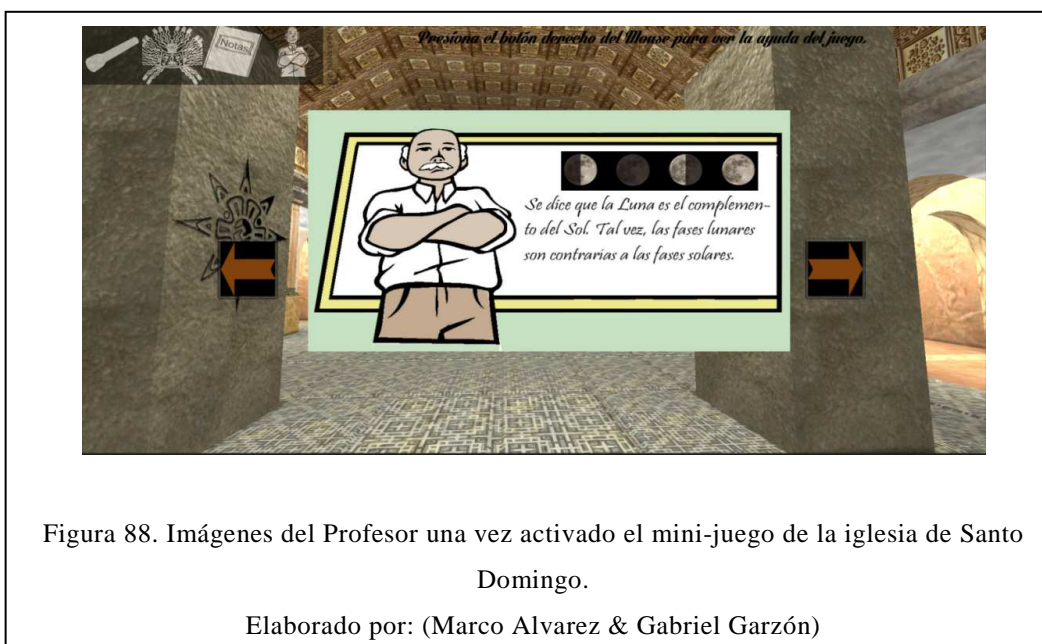
```

La forma en que actúa el script en el juego es el siguiente.

En este ejemplo el jugador se encuentra en el escenario de la iglesia de “Santo Domingo”, al momento de comenzar este nivel se puede ingresar a las herramientas y presionar en el botón de “Profesor”, estas imágenes alientan el investigar el escenario para activar el mini-juego.



Una vez activado el mini-juego, las imágenes referentes al “Profesor” cambian, estas dan una especie de ayuda al jugador para que pueda resolver el acertijo propuesto.





## CAPÍTULO 5

### PRUEBAS Y RESULTADOS

Las pruebas son acciones con las cuales un sistema y sus componentes son ejecutados en circunstancias que están previamente especificadas, estas se establecen a lo largo del proceso de desarrollo del software.

Al revisar información sobre pruebas en libros y publicaciones referentes a este tema se logró observar que existen varias que se pueden acoplar a los videojuegos, pero no totalmente; además, se verificó información en la industria la cual ayudo en gran parte a implementar las pruebas que se comentarán a continuación, Javier Gutiérrez colaborador de GameUS<sup>5</sup> y profesor de informática de la Universidad de Madrid explica como las pruebas Alfa y Beta dirigidas a testers ayudan a encontrar errores que pueden llegar a estropear la funcionalidad del juego. (Gutiérrez, 2011).

La compañía Blizzard Entertainment para colocar en el mercado uno de sus juegos, World of Warcraft, que mayor acogida ha tenido, realizo varias pruebas en diferentes Sistemas Operativos y con diferentes características de máquinas con el fin de encontrar un sistema óptimo con el cual disfrutar del juego, ya que incluye gráficos 3D y animaciones que solo en características de ciertas maquinas pueden funcionar.

Un videojuego tiene requisitos de experiencia de usuario diferentes a otros tipos de software. (Sánchez & Zea, 2012) Esto hace imprescindible el contar con un equipo humano que pruebe el juego y ajuste su experiencia modificando la dificultad, los controles, las animaciones, si es el caso los diálogos, los desafíos, etc. (Rodriguez, 2002)

---

<sup>5</sup>GameUS es una comunidad sevillana de desarrolladores de software enfocada a videojuegos que dan pautas y guías para la elaboración de videojuegos.

Sin embargo, existen otros muchos aspectos de un videojuego que se pueden probar con pruebas inmediatas. Al ver el interior de un juego quitando su interfaz gráfica, lo que se encuentra es un conjunto de funciones y estructuras de datos que no tienen una gran diferencia respecto al código de otro tipo de aplicaciones de software.

Dentro de las pruebas a realizarse en un videojuego se debe considerar los siguientes aspectos:

- Simular el comportamiento de usuario, es decir poder representar como un jugador puede reaccionar ante diferentes situaciones en un videojuego
- Verificar escenarios sin que una persona tenga que ver el juego en la pantalla, esto hace referencia a que se debe diagramar con anticipación los diferentes elementos y niveles antes de tener un resultado final.
- Lidar con las dependencias del reloj interno, es decir no depender de tiempos en los cuales se deba pasar cierto nivel o escenario, esto para que no se pierda el interés del videojuego.
- Probar que el juego siempre logra mantener el interés del jugador y de este modo no abandone el juego.

Dentro del plan de pruebas se han considerado las siguientes:

- Pruebas de funcionalidades

Proceso de revisión y verificación del sistema del software producido para revisar si se cumple con las especificaciones y que se logra el cometido

- Prueba de Sistema. (Sistema Operativo)

Buscar problemas de compatibilidad y conversión en los sistemas. Verificar y validar que el sistema se instala apropiadamente en cada cliente, bajo las siguientes condiciones:

- Instalaciones nuevas, nuevas máquinas a las que nunca se les ha instalado el software.
- Actualizar máquinas previamente instaladas con el sistema.
- Pruebas Alfa: Se invita al cliente al entorno de desarrollo, trabajando sobre un entorno controlado
- Pruebas Beta: Se desarrollan en el entorno del cliente, que se queda sólo con el producto en un entorno sin controlar.

## 5.1 Pruebas de funcionalidades

Los diferentes requerimientos que el software realiza deben tener una explicación clara y precisa de los problemas que se presentan, facilitando la generación de casos de prueba y facilitando el análisis y resolución de errores, esto para asegurar exactitud y coherencia de los requisitos del sistema.

Tabla 1. Prueba de funcionalidad No. 1

<b>Caso de prueba:</b> Configurar resolución de pantalla e iniciar el juego	<b>No. 1</b>
<b>Precondiciones:</b> El juego se inició correctamente, se ingresó al menú inicial.	
<b>Resultado esperado:</b> La configuración de pantalla de la selección se configura satisfactoriamente. Se inicia el juego.	
<b>Ejecución:</b> Se seleccionó una resolución disponible de las desplegables al iniciar el juego, se confirmó estos cambios, el sistema despliega el menú inicial, se selecciona el del menú iniciar juego y se observan las presentaciones del juego.	
<b>Resultado obtenido:</b> Se cumplió satisfactoriamente.	

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Tabla 2. Prueba de funcionalidad No. 2

<b>Caso de prueba:</b> Ingresar al juego y saltarse las presentaciones	<b>No. 2</b>
<b>Precondiciones:</b> El juego se inició correctamente, se ingresó al menú inicial.	
<b>Resultado esperado:</b> El juego inicia adecuadamente	
<b>Ejecución:</b> Se selecciona del menú inicial iniciar juego, comienzan las presentaciones o animaciones, se presiona la tecla esc, se ingresa a la jugabilidad del sistema.	
<b>Resultado obtenido:</b> Se cumplió satisfactoriamente.	

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Tabla 3. Prueba de funcionalidad No. 3

<b>Caso de prueba:</b> Activar mecanismo de pilares en nivel de la iglesia	<b>No. 3</b>
<b>Precondiciones:</b> El juego se encuentra en progreso	
<b>Resultado esperado:</b> Se activa el mecanismo que se encuentra en el piso, se levantan los pilares	
<b>Ejecución:</b> Se busca el mecanismo ubicado en el piso de la iglesia, se levantan los pilares, se verifica las herramientas, al dar clic derecho del mouse se presenta información sobre el mecanismo de los pilares.	
<b>Resultado obtenido:</b> Se cumplió satisfactoriamente.	

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Tabla 4. Prueba de funcionalidad No. 4

<b>Caso de prueba:</b> Activar mecanismo de la cámara secreta bajo la iglesia	<b>No. 4</b>
<b>Precondiciones:</b> El juego se encuentra en progreso	
<b>Resultado esperado:</b> Se activa el mecanismo que se encuentra en la pared de la cámara secreta, el ídolo de la pared cae para recogerlo.	
<b>Ejecución:</b> Se ingresa a la cámara secreta, se verifica las imágenes del mecanismo de la pared, se da clic derecho del mouse sobre cada imagen para verificar la información de ayuda para resolver el acertijo, cada imagen retrocede como si fuera un botón, el sol inca cae de la pared, se recoge el sol inca, el sol se almacena en las herramientas del jugador	
<b>Resultado obtenido:</b> Se cumplió satisfactoriamente.	

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Tabla 5. Prueba de funcionalidad No. 5

<b>Caso de prueba:</b> Mover personaje en primera persona por el nivel de la montaña y buscar el templo inca	<b>No. 5</b>
<b>Precondiciones:</b> El juego se encuentra en progreso	
<b>Resultado esperado:</b> Se realiza movimientos en el escenario de la montaña	
<b>Ejecución:</b> Se presiona las teclas de movimiento, se busca e ingresa al templo inca	
<b>Resultado obtenido:</b> Se cumplió satisfactoriamente.	

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Tabla 6. Prueba de funcionalidad No. 6

<b>Caso de prueba:</b> Iniciar nuevo juego	<b>No. 6</b>
<b>Precondiciones:</b> El juego se inició completamente, se cargó el menú inicial.	
<b>Resultado esperado:</b> El primer escenario se cargó completamente, se inició la interacción entre el sistema, jugador y escenario.	
<b>Ejecución:</b> Se seleccionó la opción iniciar desde el menú inicial, se presenta el mensaje de cargando, se inicia la animación de inicio, el jugador puede interactuar en el escenario.	
<b>Resultado obtenido:</b> Se cumplió satisfactoriamente.	

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Tabla 7. Prueba de funcionalidad No. 7

<b>Caso de prueba:</b> Salir del juego	<b>No. 7</b>
<b>Precondiciones:</b> El juego se encuentra en ejecución, se despliega el menú final del juego	
<b>Resultado esperado:</b> Se termina la ejecución del juego luego de la animación final.	
<b>Ejecución:</b> Se seleccionó la opción salir del menú final del juego, el sistema finaliza y regresa al sistema operativo.	
<b>Resultado obtenido:</b> Se cumplió satisfactoriamente.	

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Tabla 8. Prueba de funcionalidad No. 8

<b>Caso de prueba:</b> Reiniciar el juego una vez concluido	<b>No. 8</b>
<b>Precondiciones:</b> El juego se encuentra en ejecución, se despliega el menú final del juego	
<b>Resultado esperado:</b> Se termina la ejecución del juego luego de la animación final.	
<b>Ejecución:</b> Se seleccionó la opción volver a jugar del menú final del juego, se reinicia el juego desde la animación inicial.	
<b>Resultado obtenido:</b> Se cumplió satisfactoriamente.	

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Tabla 9. Prueba de funcionalidad No. 9

<b>Caso de prueba:</b> Forzar salida del juego en caso de error	<b>No. 9</b>
<b>Precondiciones:</b> El juego se encuentra en ejecución.	
<b>Resultado esperado:</b> Se termina la ejecución del juego.	
<b>Ejecución:</b> El juego se encuentra en ejecución, se presenta algún error en la ejecución, se presiona la combinación de teclas alt+f4, se sale del juego y se ingresa al sistema operativo.	
<b>Resultado obtenido:</b> Se cumplió satisfactoriamente.	

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Tabla 10. Prueba de funcionalidad No. 10

<b>Caso de prueba:</b> Seleccionar pilares	<b>No. 10</b>
<b>Precondiciones:</b> El juego se encuentra en ejecución, se ejecuta el mecanismo de pilares del primer nivel.	
<b>Resultado esperado:</b> Al seleccionar los pilares estos bajan según se dé clic sobre las imágenes del sol y luna.	
<b>Ejecución:</b> Se ingresa al primer nivel, se activa el mecanismo, se acerca a los pilares y se da clic sobre ellos, bajan los pilares, si se presiona la combinación correcta se da paso al otro nivel.	
<b>Resultado obtenido:</b> Se cumplió satisfactoriamente.	

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Tabla 11. Prueba de funcionalidad No. 11

<b>Caso de prueba:</b> Seleccionar pilares	<b>No. 11</b>
<b>Precondiciones:</b> El juego se encuentra en ejecución, se ejecuta el mecanismo de pilares del primer nivel.	
<b>Resultado esperado:</b> Al seleccionar los pilares estos bajan según se dé clic sobre las imágenes del sol y luna.	
<b>Ejecución:</b> Se ingresa al primer nivel, se activa el mecanismo, se acerca a los pilares y se da clic sobre ellos, bajan los pilares, si se presiona la combinación correcta se da paso al otro nivel.	
<b>Resultado obtenido:</b> Se cumplió satisfactoriamente.	

Elaborado por: (Marco Alvarez & Gabriel Garzón)



Tabla 12. Prueba de funcionalidad No. 12

<b>Caso de prueba:</b> Iniciar escenario	<b>No. 12</b>
<b>Precondiciones:</b> Finalizar un nivel o escenario	
<b>Resultado esperado:</b> Se inicia correctamente el siguiente nivel.	
<b>Ejecución:</b> El nivel termina y es retirado, se muestra el siguiente nivel.	
<b>Resultado obtenido:</b> Se cumplió satisfactoriamente.	

Elaborado por: (Marco Alvarez & Gabriel Garzón)

Tabla 13. Prueba de funcionalidad No. 13

<b>Caso de prueba:</b> Recoger sol inca	<b>No. 13</b>
<b>Precondiciones:</b> Resolver el acertijo del segundo nivel	
<b>Resultado esperado:</b> Se recoge el sol y se añade a las herramientas el sol inca	
<b>Ejecución:</b> Se resuelve el acertijo de la historia del primer inca, el sol de la pared cae, recogemos el sol, el objeto desaparece de la pared.	
<b>Resultado obtenido:</b> Se cumplió satisfactoriamente.	

Elaborado por: (Marco Alvarez & Gabriel Garzón)

## **5.2 Pruebas de sistema (Sistema Operativo)**

Se basan en el acoplamiento del hardware y software en el sistema que dispone el computador en el que se va a realizar la instalación de la aplicación. Para realizar las pruebas de rendimiento se instaló el videojuego en dos computadores, un computador con los requerimientos de hardware y software básico y otro con los requerimientos de hardware y software recomendados (ver capítulo 1.6).

El FPS (frames por segundo) se refiere a la velocidad en la que se puede reproducir fotogramas o imágenes en un segundo, si estas imágenes juntas forman una secuencia (como un personaje caminando o corriendo) al reproducirlas a una velocidad apropiada genera la sensación de movimiento para el espectador. Una velocidad apropiada para el FPS está entre 25 – 30. Se tomó como parámetro de referencia los fps para determinar el rendimiento del videojuego.

Los parámetros de configuración son los siguientes:

### **Resolución de pantalla para pruebas de Sistema Operativo:**

Que puede ser:

- 1024 \* 768
- 800 \* 600

### **Detalle gráfico para pruebas de Sistema Operativo:**

Que puede ser:

- Alto: con sombras dinámicas, mapeo de luz completo, texturas de 512 \* 512 pixeles de definición.
- Medio: sin sombras dinámicas, mapeo de luz ligero, texturas de 256 \* 256 pixeles de definición.

- Bajo: sin sombras dinámicas, sin mapeo de luz, texturas de 128 \* 128 pixeles de definición.

El computador de requerimientos básicos tiene las siguientes características:

- Procesador Intel Pentium IV 2,0 GHz
- Memoria RAM 1GB
- Disco duro de 160 GB con 1GB de espacio libre
- Tarjeta de sonido Sound Blaster Audigy 16 bits
- Tarjeta de video Gforce 7600 GT de 256 MB
- Sistema Operativo Windows XP SP3

Los resultados obtenidos fueron los siguientes:

Tabla 14. Resultados pruebas de Sistema Operativo				
Configuración	FPSmin	FPSmax	FPSprom	Resultado
Resolución: 1024 * 768 Detalle: alto	8	16	12	No pasa la prueba
Resolución: 800 * 600 Detalle: alto	10	19	14,5	Pasa la prueba pero con errores de continuidad en las animaciones
Resolución: 1024 * 768 Detalle: medio	14	22	18	Pasa la prueba pero con errores de continuidad en las animaciones
Resolución: 800 * 600 Detalle: medio	17	26	21,5	Pasa la prueba
Resolución: 1024 * 768 Detalle: bajo	22	29	25,5	Pasa la prueba
Resolución: 800 * 600 Detalle: bajo	23	31	27	Pasa la prueba
Elaborado por: (Marco Alvarez & Gabriel Garzón)				

El computador de requerimientos recomendados tiene las siguientes características:

- Procesador Intel Pentium Dual Core 2,8 GHz
- Memoria RAM 2GB
- Disco duro de 250 GB
- Tarjeta de sonido Sound Blaster Live 24 bits
- Tarjeta de video Gforce 8600 GT de 512 MB
- Sistema Operativo Windows XP SP3

Los resultados obtenidos fueron los siguientes:

Tabla 15. Resultados pruebas de Sistema Operativo

Configuración	FPS min	FPS max	FPS prom	Resultado
Resolución: 1024 * 768 Detalle: alto	20	30	25	Pasa la prueba
Resolución: 800 * 600 Detalle: alto	21	31	26	No pasa la prueba
Resolución: 1024 * 768 Detalle: medio	24	32	28	Pasa la prueba
Resolución: 800 * 600 Detalle: medio	25	32	28,5	No pasa la prueba
Resolución: 1024 * 768 Detalle: bajo	28	32	30	Pasa la prueba
Resolución: 800 * 600 Detalle: bajo	29	32	30,5	Pasa la prueba

Elaborado por: (Marco Alvarez & Gabriel Garzón)

### **5.3 Pruebas Alfa y Beta**

Pruebas que sirven para descubrir errores que sólo el usuario final puede descubrir, dependiendo de las características de hardware y por medio de calificaciones otorgadas se evaluará los siguientes aspectos:

- Gráficos 3D
- Interfaz de usuario: menús
- Sonido: música y efectos sonoros
- Jugabilidad: comodidad del jugador al manejar el personaje
- Historia: trama del videojuego

Se solicitó a los usuarios que realicen su calificación en una escala de 0 a 100.

#### **5.3.1 Pruebas Alfa.**

Las pruebas Alfa se efectuaron sobre el equipo donde se desarrolló el videojuego. En primer lugar y para simular, en una primera instancia el ambiente del jugador, se generó el proyecto completo, se instaló tal cual debería permanecer en el ambiente final. Las características del equipo de desarrollo son las siguientes:

Tabla 16. Características PC

<b>PARTE</b>	<b>VALOR</b>
Marca y modelo del equipo	Clon de escritorio
Procesador	Intel Dual Core
Memoria RAM	6 GB RAM DDRII
Tarjeta madre	Intel 102G
Disco duro	Western Digital 300GB IDE
Video	MSI Geforce 8500GT 512MB
Sonido	Creative Soundblaster live 24bits
Sistema operativo	Windows 7

Elaborado por: (Marco Alvarez & Gabriel Garzón)

### Configuración del videojuego

Tabla 17. Características resolución de pantalla

<b>PARTE</b>	<b>VALOR</b>
Resolución	1024 x768 pixeles
Detalle	Alto

Elaborado por: (Marco Alvarez & Gabriel Garzón)

### Datos del jugador

Tabla 18. Datos del jugador

<b>DATOS</b>	<b>VALOR</b>
Nombre	Luis Minas
Edad	28 años
Experiencia como jugador	Intermedio

Elaborado por: (Marco Alvarez & Gabriel Garzón)

### **Conclusiones de las pruebas Alfa:**

Las conclusiones dadas a continuación son en base a una conversación realizada después de terminar el juego y es tomada como una apreciación personal.

- La fluidez de gráficos es buena. Los cuerpos tridimensionales se evalúan en un nivel medio, ya que se utiliza una cantidad de polígonos limitada.
- Las texturas de los ambientes, sobre todo abiertos, requieren mayor resolución ya que al aproximarse y observar los objetos detenidamente pierden el detalle y se denota pixelación.
- La animación inicial, final y efectos es baja.
- Agregar movimientos más variados traería más realismo al juego.
- No se presentan secuencias de video utilizando los elementos 3D del juego, lo que podría ayudar a conseguir una historia y desarrollo de la trama más atractiva e interesante.

#### **5.3.2 Pruebas Beta.**

Para realizar las pruebas Beta, se instaló el videojuego en los equipos de los jugadores seleccionados. Se consideró un grupo de 10 jugadores comprendidos entre los 8 y 15 años, se realizó la prueba en para ver cuáles eran las apreciaciones que tenían del juego.

#### **Prueba Beta #1**

El ambiente para la prueba Beta #1 es el siguiente:

## Características

Tabla 19. Características PC prueba Beta No. 1

PARTE	VALOR
Marca y modelo del equipo	HP Pavilion dv9548us Entertainment Notebook PC
Procesador	Intel Core 2 Duo 1,67GHz
Memoria RAM	4 GB RAM DDRII
Tarjeta madre	Incorporada
Disco duro	Fujitsu MHW2120 250GB SATA
Video	nVidia Geforce 8600M GS PCI-Express 512MB
Sonido	Realtek High Definition Audio
Control	Teclado standard PS/2, mouse Genius
Sistema operativo	Windows Vista Home Premium

Elaborado por: (Marco Alvarez & Gabriel Garzón)

## Configuración del videojuego

Tabla 20. Resolución pantalla, prueba Beta No. 1

PARTE	VALOR
Resolución	1024 x768 pixeles
Detalle	Alto

Elaborado por: (Marco Alvarez & Gabriel Garzón)

## Datos del jugador

Tabla 21. Dato jugador prueba Beta No. 1

DATOS	VALOR
Nombre	Paula Balarezo
Edad	12 años
Experiencia como jugador	Principiante

Elaborado por: (Marco Alvarez & Gabriel Garzón)



## Apreciación:

Tabla 22. Apreciación prueba Beta No. 1

ASPECTOS	CALIFICACIÓN
Gráficos 3D	75
Interfaz de usuario	60
Sonido	65
Jugabilidad	70
Historia	80
CALIFICACIÓN GENERAL	70

Elaborado por: (Marco Alvarez & Gabriel Garzón)

## Prueba Beta #2

### Características

Tabla 23. Características PC prueba Beta No. 2

PARTE	VALOR
Marca y modelo del equipo	Clon
Procesador	Intel Pentium 4 2,4GHz
Memoria RAM	1GB RAM
Tarjeta madre	Intel 845GLLY
Disco duro	Seagate Barracuda 5400rpm IDE
Video	XFX Geforce 7600GT 256MB
Sonido	Realtek High Definition Audio
Control	Teclado estándar PS/2, mouse estándar PS/2
Sistema operativo	Windows XP SP2

Elaborado por: (Marco Alvarez & Gabriel Garzón)

## Configuración del videojuego

Tabla 24. Resolución pantalla, prueba Beta No. 2

PARTE	VALOR
Resolución	800 x 600 pixeles
Detalle	bajo

Elaborado por: (Marco Alvarez & Gabriel Garzón)

## Datos del jugador

Tabla 25. Dato jugador prueba Beta No. 2

DATOS	VALOR
Nombre	Pablo Gallegos
Edad	15
Experiencia como jugador	Media

Elaborado por: (Marco Alvarez & Gabriel Garzón)

## Apreciación:

Tabla 26. Apreciación prueba Beta No. 2

ASPECTOS	CALIFICACIÓN
Gráficos 3D	60
Interfaz de usuario	55
Sonido	70
Jugabilidad	65
Historia	45
CALIFICACIÓN GENERAL	59

Elaborado por: (Marco Alvarez & Gabriel Garzón)

## Prueba Beta #3

### Características

Tabla 27. Características PC prueba Beta No. 3

PARTE	VALOR
Marca y modelo del equipo	Clon
Procesador	AMD Athlon 64 X2
Memoria RAM	4 GB DDRII
Tarjeta madre	MSI K9N6SGMV
Disco duro	Samsung 7200rpm 500GB
Video	Geforce LSI 512MB
Sonido	Realtek High Definition Audio
Control	Teclado PS/2, mouse PS/2
Sistema operativo	Windows XP SP2

Elaborado por: (Marco Alvarez & Gabriel Garzón)

## Configuración del videojuego

Tabla 28. Resolución pantalla, prueba Beta No. 3

PARTE	VALOR
Resolución	1024x768
Detalle	Alto

Elaborado por: (Marco Alvarez & Gabriel Garzón)

## Datos del jugador

Tabla 29. Dato jugador prueba Beta No. 3

DATOS	VALOR
Nombre	Matías Salazar
Edad	13 años
Experiencia como jugador	Intermedio

Elaborado por: (Marco Alvarez & Gabriel Garzón)

## Apreciación:

Tabla 30. Apreciación prueba Beta No. 3

ASPECTOS	CALIFICACIÓN
Gráficos 3D	70
Interfaz de usuario	45
Sonido	75
Jugabilidad	80
Historia	90
CALIFICACIÓN GENERAL	72

Elaborado por: (Marco Alvarez & Gabriel Garzón)

## CONCLUSIONES

- Se logró crear un prototipo de videojuego en el cual, de manera didáctica, se dan a conocer algunos hechos históricos importantes de Quito y del Ecuador.
- El prototipo de videojuego no pretende ser un sustituto de los elementos típicos en el proceso de enseñanza-aprendizaje, sino tomar el papel de un complemento para la misma.
- Mediante las dinámicas del videojuego se proporciona una forma novedosa de integración entre un software de entretenimiento y la materia Historia.
- Los diferentes mini-juegos que integran el prototipo no deben conllevar una interacción fastidiosa o molesta para el jugador.
- Dados los resultados de las pruebas realizadas se considera que el videojuego puede ser considerado como un elemento de apoyo en el proceso de enseñanza-aprendizaje.
- El videojuego propone una interacción con el usuario mucho más amplia que en una aplicación normal, invitándolo a la exploración de los diferentes escenarios y la resolución de los diferentes mini-juegos.
- El motor de juegos Unity es una herramienta que brinda facilidades para el desarrollo de videojuegos, su IDE y editor de scripts “MonoDevelop” brindan un excelente ambiente de trabajo para el manejo de los diferentes componentes que actúan en el videojuego.
- El uso de software propietario como Autodesk – Maya no es obligatorio, en el mercado existen otras opciones como programas con licencias

gratuitas tal es el caso de Blender, dicho programa realiza perfectamente el proceso realizado en Maya.

- El desarrollo de un videojuego puede no conllevar limitantes o reglas como en la creación de módulos que realicen una determinada tarea en cualquier otra aplicación, las reglas que maneja un videojuego son enteramente creadas por el equipo creativo del mismo.
- Los Ingenieros en Sistemas que integran equipos de trabajo en el desarrollo de videojuegos pueden llegar a obtener y comprender puntos de vista no tan técnicos y mucho más creativos.
- Los encargados del desarrollo de scripts en un videojuego creado en Unity tienen dos alternativas referentes al lenguaje de programación a utilizar, C# y Javascript, el uso de uno debe ser considerado una parte importante en el flujo de trabajo dado que la comunicación entre lenguajes no es permitido.
- La práctica de metodologías ágiles en el desarrollo de videojuegos significa un uso adecuado del tiempo que se maneja, el videojuego puede ser dividido en un número de entregables manejables para un grupo pequeño de desarrollo.
- Los entregables que propone Scrum, en el desarrollo de videojuegos pueden ser de diferente índole como por ejemplo: scripts, modelos 3D, texturizados de dichos modelos, escenarios creados, animaciones, etc.
- Los videojuegos pueden ser considerados herramientas que ayudan al desarrollo de diferentes estímulos como por ejemplo entre el ojo y la mano siendo a su vez eficaces instrumentos de rehabilitación en terapias psicomotrices...

- El desarrollo de este proyecto resultó, en algunas etapas, un trabajo extenuante y muy agotador pero en contraste se puede decir que el aprendizaje resultante ha sido muy significativo para el equipo de desarrollo.
- Los Ingenieros en Sistemas no especializados en el desarrollo de videojuegos, de decidir incursionar en este ámbito pueden experimentar un incremento de sus destrezas creativas.

## RECOMENDACIONES

- Hay que tener en consideración el no recargar los recursos tanto visuales como auditivos en el videojuego, ya que estos pueden causar graves complicaciones al ejecutar el mismo, es conveniente seguir un plan o guión para delimitar todo lo que se va a realizar.
- Esté videojuego es un prototipo, por lo cual se recomienda un estudio del mismo como base para trabajos subsiguientes o complementarios.
- Como prototipo, en esté videojuego se pueden modificar muchas apartados e integrar nuevas funcionalidades con el objetivo de tener una mayor interacción entre el jugador y el software.
- Se recomienda que el tutor de la materia Historia complemente el software a sus clases y que además, proporcione información adicional a sus alumnos.
- Si se pretende cambiar la interacción entre el jugador y el videojuego se recomienda en todo sentido que antes de hacerlos se piense en el usuario final, si esta interacción es difícil o fastidiosa el usuario perderá interés y no se alcanzará el objetivo principal del proyecto.
- Se recomienda a todo iniciador o emprendedor que quiera profesionalizarse en esta industria el aprender el lenguaje de programación C++ dado que la mayoría de juegos profesionales son desarrollados en motores de juegos que manejan dicho lenguaje.
- Programas como Blender o Gimp, que cuentan con licencias gratuitas de uso, son alternativas excelentes para equipos de desarrollo que quieren dedicarse completamente al desarrollo de videojuegos.

- Para el desarrollo de videojuegos se recomienda tener un amplio conocimiento técnico, pero también una buena visión creativa, el leer libros de fantasía o conocer historias acordes a este tema puede ser una gran ayuda, esto brindará al desarrollador ideas que se plasmarán en soluciones de forma que representen una mayor jugabilidad.
- Se recomienda aprender los dos tipos de lenguajes que son mayormente utilizados con el motor de juegos Unity, JavaScript y C#, es vital, en el flujo de trabajo, el escoger correctamente en que lenguaje se desarrollará un videojuego.
- Las reuniones pactadas por la metodología Scrum para la inspección de entregables pueden ser realizadas semanalmente, también se recomienda que tengan estas personalmente, de no ser posible dichas reuniones personales, es recomendable manejarlas de manera virtual mediante video-sesiones.



## LISTA DE REFERENCIAS

- *La noche interminable. Investigación.* (5 de septiembre de 2010). Recuperado el 10 de agosto de 2013, de <http://sandiegoysanvicente.com/narracine/proyectos/proyecto-grupo02-20a/grupo-a2-investigacion-2/>
- *Clasificación de videojuegos.* (02 de mayo de 2010, 1-2). Recuperado el 20 de agosto de 2013, de <http://edutec.rediris.es/Revelec2/revelec15/anexo.pdf>.
- *La industria de los videojuegos.* (13 de septiembre de 2011). Recuperado el 25 de agosto de 2013, de <http://www.pymempresario.com/2012/02/la-industria-de-los-videojuegos-una-oportunidad-de-crecimiento-para-mexico/>
- Acerenza, N., & Mesa, G. (5 de julio de 2009). *Una Metodología para Desarrollo deVideojuegos.* Recuperado el 13 de septiembre de 2012, de [http://www.fing.edu.uy/inco/grupos/gris/wiki/uploads/Proceedings/ASSE\\_2009\\_16.pdf](http://www.fing.edu.uy/inco/grupos/gris/wiki/uploads/Proceedings/ASSE_2009_16.pdf).
- Autodesk. (20 de julio de 2012). *Software completo de animación en 3D.* Recuperado el 15 de agosto de 2013, de <http://www.autodesk.es/products>
- Estrella, M. (01 de diciembre de 2007). *Scrum en Español.* Recuperado el 20 de septiembre de 2013, de <http://scrumenespanol.blogspot.com/2007/12/scrum-en-pocas-palabras.html>
- Gil, A., & Vida, T. (2007). *Los Videojuegos.* Barcelona, España: UOC.

- González, D. (20 de marzo de 1995). Recuperado el 28 de septiembre de 2013, de <http://www.advertainmen.com/2007/12/cmo-crear-un-guin-de-videojuego-y-que.html>
- Gonzalez, D. (16 de abril de 2009). La importancia del guión. *PC ACTUAL*, 10-12.
- Gutiérrez, J. (2011). Videojuegos, bugs y cambios. *Informatica MAdrid*, 10-12.
- Hernández, R. (12 de Junio de 2010). *www.yumpu.com*. Recuperado el 15 de agosto de 2013, de <https://www.yumpu.com/es/document/view/14652191/columna-de-desarrollo-de-video-juegos-willy-net>
- itween. (20 de marzo de 2012). *itween.pixelplacement*. Recuperado el 11 de octubre de 2013, de <http://itween.pixelplacement.com/index.php>
- Lacalle, A. (10 de julio de 2006). *Alberto Lacalle, Diseño de software*. Recuperado el 16 de septiembre de 2013, de [http://albertolacalle.com/hci\\_prototipos.htm](http://albertolacalle.com/hci_prototipos.htm):  
[http://albertolacalle.com/hci\\_prototipos.htm](http://albertolacalle.com/hci_prototipos.htm)
- Lacassa, P. (2011). *Los Videojuegos, Aprender en mundos virtuales y reales*. Madrid: MORATA S.L.
- Letelier, P. (25 de abril de 2006). *Métodologías ágiles para el desarrollo de software*. Recuperado el 15 de febrero de 2014, de [www.cyta.com.ar/ta0502/v5n2a1.htm](http://www.cyta.com.ar/ta0502/v5n2a1.htm)

- Melián, M. (2009). *La calidad del software y su medida*. Madrid.
- Pincay Ponce, J. (14 de septiembre de 2008, 5-8). Prototipos de videojuegos para PC programados con Visual C++.NET. Manta, Manabi, Ecuador.
- Radulescu, M. (12 de mayo de 2012). *Lenguaje Cinematográfico*. Recuperado el 10 de marzo de 2013, de <http://lc-lenguajecinematografico.blogspot.com/2014/05/la-narrativa-detras-del-guion.html>
- Rodriguez, E. (2002). *Jovenes y Videojuegos*. Madrid: Injuve.
- Sánchez, G., & Zea, P. (abril de 2012). Jugabilidad como Calidad de la Experiencia del Jugador. *Publicación Mensual, univervidad Castilla*, 23-25.
- TechnologiesUnity. (18 de febrero de 2012). *Unity3D*. Recuperado el 10 de septiembre de 2013, de <http://spanish.unity3d.com/create-games/>
- UnityDocumentation. (23 de enero de 2012). *Unity 3D*. Recuperado el 27 de septiembre de 2013, de <http://docs.unity3d.com/Documentation/ScriptReference/GameObject.Find.html>
- Villalobos, J., & Candil, D. (17 de enero de 2009). *Vida Extra*. Recuperado el 16 de agosto de 2013, de <http://www.vidaextra.com/cultura/la-importancia-del-guion-en-los-videojuegos-2>

## Anexo 1. Guión del juego

“El Imperio Inca domino por muchos siglos gran parte de lo que ahora se llama América del Sur, entre sus grandes hazañas están sus imponentes construcciones, sus grandes artes incaicas y por sobre todo la organización que ellos manejaban como una forma de vida, gracias a ello, en el Imperio no existía la pobreza pero al mismo tiempo tampoco existía la pereza.

Según cuentan las narraciones y leyendas, se dice que los Incas eran descendientes del Dios Sol Inti porque este envió a su hijo Ayar Manco con su esposa Mama Ocllo para que pudieran civilizar a los humanos, al haber fundado los cimientos para el imperio Ayar Manco se convierte en Manco Cápac, el primer Inca y gobernante de lo que en el futuro se conocería como Imperio Incaico.

Él junto a su compañera enseñan a los humanos como pescar, cultivar, construir y elaborar sus vestimentas realizando la tarea que les fue encomendada, los emperadores que les siguieron al ser descendientes del hijo del Dios Inti afirmaban el parentesco que tenían con el mismo.

Los descendientes de Manco Cápac fueron bastante, pero esta historia se centrará en el último de ellos, Atahualpa. El último emperador era hijo de HuaynaCápac y hermano de Huáscar, al morir su padre el reino queda dividido en dos, los territorios del norte para Atahualpa y los del sur para Huáscar, esto llevo a una guerra entre hermanos por el control de todo el territorio Inca y que al final también llevaría a la destrucción del Imperio.

Al haber ganado una guerra más que sangrienta entre los ejércitos de los hijos de HuaynaCápac, Atahualpa se proponía ir a la ciudad sagrada de Cuzco para ser coronado como nuevo Emperador, pero para mala suerte del nuevo gobernante, forasteros llegan de tierras lejanas sin muy buenas intenciones, eran conquistadores españoles.

La mayoría de forasteros conquistadores eran gente sin ninguna preparación académica, según se cuenta, el viaje que se realizaban al nuevo continente eran cumplidos por personas que no tenían otra opción, eran obligados a embarcarse en una aventura sin seguridad de regresar con vida, por lo cual en muchos casos la tripulación estaba formada de gente despiadada, cruel y muchas veces malévola.

Por el año de 1533 llegan a tierras latinoamericanas una expedición liderada por el español Francisco Pizarro, los cuales se encontraron con la civilización Incaica y también con sus grandes riquezas en oro, los hombres al mando de Pizarro eran muy pocos en comparación al ejército Incaico, incluso con sus armas más avanzadas sería muy difícil el poder someter a un grupo tan grande pero para suerte del español en esas mismas fechas se culminó una gran batalla dentro del Imperio, la que tuvieron Huáscar y Atahualpa, por lo cual existían dentro del pueblo Inca aún seguidores del ya derrotado Huáscar contrarios al nuevo emperador, Pizarro aprovecho este hecho para poder formular un plan para secuestrar a la cabeza del Imperio Inca.

Principalmente los españoles se presentaron como gente de paz e hicieron creer que venían con una misión de establecer relaciones entre su nación y los Incas, por su parte el pueblo creía que los forasteros eran dioses ya que montaban caballos y portaban armaduras brillantes, estos objetos nunca habían sido vistos por ningún ojo Inca antes, pero el Emperador había descubierto que al contrario de dioses, los forasteros, eran igual de humanos que él. Los planes del Emperador al recibir a los españoles era ganar su confianza y luego derrotarlos de esta manera unificaría a su pueblo al mostrar que él podría vencer a seres “divinos”. Los españoles siguieron el guion del Emperador, aprendieron su lengua al igual que ellos enseñaron la suya a los Incas, aprendieron de sus costumbres y en especial de la forma en que ellos se defendían, Pizarro fue muy cauteloso en todo momento al no presentar el verdadero potencial de sus armas, esto lo llevo a entablar una estrecha “amistad” con Atahualpa al punto que lo convenció de ir a un encuentro en Cajamarca.

Los españoles convencieron al Inca de no llevar guerreros en su procesión hasta el lugar del encuentro, cosa que Atahualpa aceptó a medias, llevo camuflados guerreros entre sus sirvientes ya que esta sería la oportunidad perfecta de capturar a los forasteros y mostrar su poderío. Al llegar a Cajamarca los Incas fueron sitiados y acorralados, los cañones y armas españolas, junto con traidores Incas, asustaron a los combatientes y este encuentro terminó con la captura de Atahualpa.

De aquí surge una nueva leyenda que fascina a aquel que la escucha, se dice que Atahualpa fue asesinado por los españoles pero antes ellos engañan nuevamente al Emperador diciéndole que si el concedía un pago por su vida los conquistadores se irían de sus tierras y no regresarían. Atahualpa acepta este trato y manda a que en el Imperio se comience a reunir los pedidos de los españoles en oro. Poco a poco las cargas llegaban donde los conquistadores pero el emperador se dio cuenta que los forasteros entre más oro recibían, más querían, al deducir que al completar el pedido su vida llegaría a su fin envía un mensaje a uno de sus Generales, Rumiñahui, el cual manejaba los ejércitos de Quito al norte del Imperio, “no completes el pedido de los invasores” fueron sus órdenes, de esta manera Atahualpa creyó haber ganado mucho más tiempo para dárselos a su ejército con el objetivo de poder rescatarlo. Rumiñahui realiza el mandato de su Emperador y Atahualpa convence a los españoles que las mayores riquezas que pudieran obtener eran las que estaban al norte de Imperio, las que no llegarían hasta que él no esté libre, esto provocó la furia de los españoles y terminan con la vida del Emperador, reúnen a sus tropas y marchan al norte a encontrar los tesoros que el General Rumiñahui tenía. Éste antes de que los enemigos llegaran incendia la ciudad de Quito y junto a sus tropas toman rumbo a los Llanganates –que significa Cerro Hermoso- y es aquí donde supuestamente esconde el tesoro y parte a encontrarse con las tropas enemigas, Rumiñahui es vencido y torturado pero en ningún momento, hasta su muerte, revela el paradero del tesoro, incluso hasta el día de hoy nadie sabe su paradero...”

### **Primera parte del guión**

AFUERAS DE QUITO – 05 DE NOVIEMBRE 2012 – 6PM

Dentro de una excavación cerca de la ciudad de Quito, se encuentran trabajando jóvenes, estudiantes en un descubrimiento de unas ruinas aparentemente de la cultura Incaica, dentro de estas construcciones se puede divisar a las personas trabajando. El personaje principal se llama LEO y es uno de los investigadores trabajando en las ruinas. Es estudiante de la Universidad Politécnica Salesiana, ha sido seleccionado para trabajar en la investigación de estas ruinas.

### **LEO (PENSAMIENTO)**

Así terminaba la historia que mi abuelo me contaba sobre el tesoro de Atahualpa, al comenzar a trabajar aquí siempre me acuerdo de ella, y como él siempre me pedía que la recordara “LEO lo que te cuento es parte de tu historia y la mía” decía, creo que eso me llevo a estudiar mi carrera de arqueología. Ahora estoy trabajando en un proyecto de la universidad, no hace mucho se encontraron muchos vestigios de asentamientos Incaicos cerca de la ciudad de Quito, en una de las “canteras” lugares de donde extraen minerales para la fabricación de materiales para la construcción, estos vestigios son algo nuevo incluso para los más adelantados estudios sobre el Imperio Inca, no se sabe a ciencia cierta que hacen estas construcciones tan ocultas de la superficie, pero parece ser que estas eran utilizadas como templos de adoración del Dios Inca Inti en la época de la gran Colombia, en estos momentos estoy revisando el altar, parece ser que era el único Dios adorado en esta construcción, algunos compañeros también tienen sus labores, entre ellos se encuentra DIANA una gran amiga y además, muy inteligente y Alberto, él no me cae bien cada vez que hablamos siento que hace este trabajo no por querer encontrar la verdad de nuestra historia sino por ganar fama y fortuna.

### **GUARDIA DEL TEMPLO (HABLANDO A LOS JÓVENES)**

Ya se está ocultando el sol y la planta de energía se estropeo, es mejor que vayan saliendo chicos.

Ese era uno de los guardias que acompañan a los estudiantes en la excavación, el generador que utilizan para alumbrar por la noche el sitio de investigación esta estropeado por lo tanto no podrán trabajar en la noche.

DIANA (HABLANDO A LEO)

LEO, vamos luego se hace más de noche.

LEO (LIMPIANDO UNA PARTE DE LA EXCAVACIÓN)

En un momento voy, estoy limpiando la última parte del altar, para mañana ya podremos revisarlo con mayor claridad.

DIANA (HABLANDO A LEO)

Está bien, me adelanto quiero lavarme un poco.

LEO (AÚN LIMPIANDO)

Ok, ya te alcanz.....

DIANA (GRITANDO)

¡LEO!

ALBERTO (A PUNTO DE SALIR, SE DA LA VUELTA SUBITAMENTE)

¡¿Qué ocurre?!

DIANA (HABLANDO A ALBERTO)

¡LEO se ha caído, el terreno bajo de él se ha abierto!

CAMBIO DE ESCENA.

DENTRO DEL HOYO DONDE CAYO LEO

LEO (EN EL SUELO, ACOSTADO POR LA CAIDA)

¡Auch! Me duele la espalda.



LEO comienza a revisar su cuerpo.

LEO (AUN EN EL PISO)

Aunque me duele un poco parece que nada se ha roto, trataré de levantarme.

LEO logra levantarse. El lugar donde se encuentra está muy oscuro tanto que no puede ver sus manos, busca entre su equipo la linterna que lleva en el cinturón de materiales para limpiar, comienza a revisarla tratando de encontrar el botón para encenderla.

LEO (EN PIE)

¿Dónde está? ¡Ah!, lo encontré, por favor enciende. ¡Perfecto funciona! ¿En dónde estoy? Parece ser una especie de cueva, por este lado parece que no hay salida, por el otro parece haber un camino.

DIANA (VOZ)

¡LEO! ¡Contéstame! ¡LEO!

LEO

¿DIANA?

LEO (TOMANDO SU INTERCOMUNICADOR)

¿DIANA puede oírme?!

DIANA (VOZ A TRAVEZ DEL INTERCOMUNICADOR)

¡LEO!. Gracias a Dios estas bien.

¿Estás herido?, ¿Puedes levantarte?

LEO (HABLANDO AL INTERCOMUNICADOR)

¡Si tranquila! Estoy bien.

No creo que pueda escalar hasta el templo, voy a seguir un camino que se encuentra aquí abajo.

DIANA (VOZ A TRAVEZ DEL INTERCOMUNICADOR)

¡¿Seguro?! Alberto fue por ayuda, espera un momento hasta que te lancemos una cuerda para que puedas subir.

LEO (HABLANDO AL INTERCOMUNICADOR)

¡Por el momento voy a investigar que hay aquí abajo, parece una extensión del templo!, deja que Alberto traiga ayuda si no encuentro la forma de salir de aquí regresaré a este punto para subir por la soga.

DIANA (VOZ A TRAVEZ DEL INTERCOMUNICADOR)

Está bien, ten cuidado por favor.

LEO (HABLANDO POR EL INTERCOMUNICADOR)

Ok, estamos en contacto.

LEO vuelve a poner el intercomunicador en su cinturón. Comienza a adentrarse en la cueva por el único camino que tiene, parece que es un camino artificial y no natural propia de una cueva, hay escalones que se dirigen aún más abajo.

LEO (PENSAMIENTOS)

Espero no encontrarme con un acantilado hacia el centro de la tierra. Parece que un poco más adelante acaba

LEO sigue por el camino hasta que ve una especie de entrada, en su marco se encuentran símbolos Incaicos, con la poca luz que ofrece la linterna no puede divisarlas muy bien, decide que debe ingresar a este cuarto, pasa la entrada y delante de él se encuentra otra representación del Dios Inti muy parecida a la que vio en el templo pero en su base hay algo más, una especie de objeto enrollado con una tela Incaica. LEO se acerca a este altar y comienza a revisar dicho objeto, es una especie de tela con colores muy propios de esta civilización, esta tela envuelve un objeto dentro de ella, coloca la linterna en su boca para poder utilizar las dos manos y comienza a desenvolverlo, trata de cortar los hilos alrededor de la tela que la amarra, es un cordón muy fuerte no puede romperla.

LEO (TRATANDO DE ROMPER LOS HILOS DEL OBJETO)

Increíble tantos años han de haber pasado y aun puede cumplir con su cometido. Espera tengo una navaja en mi cinturón.

Busca una especie de navaja en su cinturón de trabajo, la encuentra y la utiliza para cortar los hilos. En ese momento se escucha una especie de silbido detrás de LEO.

LEO (VOLTEANDOSE PARA VER A SUS ESPALDAS)

Extraño, creí haber oído una especie de silbido. A lo mejor sean personas que bajaron para ayudarme.

LEO camina hacia la entrada para ver si alguien se acercaba o algún rastro de luz de otra linterna pero, no encuentra nada. Sorprendido y algo asustado regresa a la labor que antes tenía y ahora con el cordón roto comienza a desenrollar la tela, al terminar se encuentra con una especie de papiro.

LEO (CON EL OBJETO EN MANO)

-¿Un mapa del tesoro? Por favor-

Guarda la tela en uno de sus bolsillos y comienza abrir el papiro para tomarlo por sus extremos y poder divisarlo mejor.

LEO (OBSERVANDO EL CONTENIDO DEL PAPIRO)

¿Qué es esto? Parecen dibujos incaicos.

El contenido del papiro era algo singular, se trataba de dibujos Incaicos, una representaba al Dios Inti, otra a la Diosa Luna. En total eran 2 representaciones, la mayor de ellas era la que representaba al Dios Inti y la otra estaba al lado de él, cada una de ellas tenía una especie de línea amarilla como un haz de luz que la conectaba. Lo que pareció más extraño para LEO eran las posiciones que tenían los dibujos, no estaban uno al lado del otro sino en diferentes posiciones del papel, como si hubieran sido pintadas al azar, además, debajo de las representaciones estaban escritas una especie de versos.

LEO escucha otra vez un ruido proveniente del camino por donde había llegado a este descubrimiento, instintivamente saca el pañuelo que estaba en su bolsillo y que antes envolvía el papiro, toma este último y lo volvió a enrollar pero en vez de cubrirlo con el pañuelo lo guarda en su guantero sujeta a su cinturón. Luego escucha unos llamados.

ALBERTO (GRITANDO)

¡LEO! ¿! Dónde estás!? ¡Joseeeee!

LEO, una vez guardado el papiro y con el pañuelo en mano se dirige a la entrada que conectaba al camino por el cual llegó, ve un haz de luz seguramente proveniente de la linterna de Alberto, por lo que contesta a los llamados de éste.

LEO (AL FRENTE DE LA ENTRADA)

¡Alberto, por aquí! ¡Sigue un poco más!

ALBERTO (APUNTANDO LA LINTERNA HACIA LEO)

¡LEO! ¡Por fin te encuentro! ¿! Estás bien!?

LEO (TAMBIEN APUNTANDO SU LINTERNA HACIA ALBERTO)

¡Sí!, Estoy bien, no te preocupes. Gracias por bajar hasta acá y buscarme.

ALBERTO (CAMINANDO HACIA LEO)

No tienes nada que agradecer, pero ¿! Qué es esto!?

Alberto noto la entrada al frente de LEO, sin preocuparse mucho de este corre a su lado y comienza a examinar la entrada.

LEO

Parecen ser como dibujos Incas aunque no estoy muy seguro. No recuerdo recordar que los Incas hicieran ese tipo de trabajos en las entradas de sus templos o viviendas.

ALBERTO (REVISANDO LA ENTRADA CON LA LUZ DE SU LINTERNA)

Tienes razón. Los Incas no tenían esta forma de realizar sus trabajos. Tal como lo había predicho, estas ruinas son muy posterior a la época de la conquista, jajaja. ¡Esto me llevara a la cúspide!

LEO (ACERCANDOSE A ALBERTO)

¿Perdón?

ALBERTO (VOLTEANDOSE PARA VER A LEO)

No nada, mejor salgamos de aquí y contemos lo que encontramos.

LEO (EN TONO SARCASTICO)

¿Encontramos?

ALBERTO (PASANDO AL LADO DE LEO EN DIRECCIÓN AL HOYO POR DONDE INGRESARON)

Si, si, vamos.

LEO (TRATANDO DE ALCANZAR A ALBERTO)

¿Acaso no quieres revisar lo que hay después de la entrada?

ALBERTO (REGRESANDO A VER A LEO CON UNA MIRADA DE CURIOSIDAD)

¿Tú ya entraste?

LEO (VIENDO FIJAMENTE A ALBERTO)

Sí, pero solo encontré otra especie de altar al Sol y este trozo de tela (levanta el trozo de tela para que Alberto pueda verla e iluminarla con la linterna), al ver que no había otra salida me propuse regresar y fue cuando te encontré.

ALBERTO (CAMBIANDO UN POCO LA MIRADA)

Ya veo. Bueno será mejor regresar mañana para revisar con mayor claridad y un mejor equipo todo lo demás.

A LEO no le supo muy bien el último comentario de Alberto pero no dijo nada, estaba un poco tranquilo por ver que su compañero aparentemente creyó su historia. Pero aún se preguntaba ¿Por qué esconder el papiro de Alberto?, ¿Acaso era egoísmo de su parte? Trato de apartar esos pensamientos de su cabeza y se dispuso seguir a su compañero.

Los dos comenzaron a caminar hasta llegar al hoyo por donde cayó LEO. Ahora del hoyo colgaba una cuerda por la cual Alberto pudo bajar. Los dos utilizaron la

cuerda para salir de ese lugar, primero fue Alberto y luego LEO. Una vez arriba LEO pudo observar a todos sus compañeros de trabajo alrededor, unos hablando con Alberto y otros tratando de ayudarlo a salir totalmente del hoyo, una vez fuera, DIANA se acerca a LEO para ayudarlo a caminar en ese momento LEO pudo ver tranquilidad en la cara de su amiga, se imaginaba que debió haber estado preocupada por los dos todo el tiempo que estuvieron ahí. De repente LEO escucha una voz familiar para él.

PROF. HURTADO (CAMINANDO A LEO Y DIANA)

¡LEO!, ¡Muchacho, es bueno verte!

El que acaba de hablar era el profesor Sergio Hurtado, él es el responsable de los jóvenes que trabajaban en las ruinas además, de ser uno de los maestros más reconocidos dentro de la facultad de Arqueología.

LEO (VIENDO AL PROFESOR)

Gracias profesor, pienso lo mismo.

PROF. HURTADO (VIENDO A LEO)

Bueno, bueno, dejemos de hablar. Anda y que te revisen para ver si está todo bien yo iré pronto a.....

De repente el profesor es interrumpido por Alberto.

ALBERTO (DETRÁS DEL PROF. HURTADO)

¡Profesor!. ¡Tengo buenas noticias! Debajo del hoyo por donde salimos hay una especie de camino subterráneo que conduce a otro altar del dios Sol. Me imagino que pueden haber muchas más caminos además, de ese debajo de todo el templo. Por ende le pido que me deje liderar un equipo para comenzar a investigar acerca de esto.

PROF. HURTADO (GIRANDO HACIA ALBERTO)

¡Alberto, eso se puede hablar más tarde!, primero hay que asegurarnos que LEO no está lastimado.

ALBERTO

Pero.. ¡Profesor!.

PROF. HURTADO

¡Alberto!

ALBERTO (MIRANDO AL PROFESOR)

De acuerdo. Luego lo hablaremos

Dicho esto Alberto dio media vuelta dirigiéndose a la salida del templo, sus últimas palabras parecieron que tenían enojo en su interior. El profesor regreso a ver a sus otros dos estudiantes.

PROF. HURTADO (VIENDO A LEO Y DIANA)

Ok, DIANA. Por favor acompaña a LEO, que lo chequeen los doctores y que se aseguren que no está lastimado de gravedad. Y tú LEO, descansa un poco.

DIANA (MIRANDO AL PROF. HURTADO)

No se preocupe profesor. Enseguida lo llevo a que lo revisen.

LEO (MIRANDO AL PROFESOR)

No creo que necesite que me revisen, en realidad no me duele nada. Solo un poco la espalda por la caída.



PROF HURTADO (VIENDO A LEO Y DIANA)

Igual, tenemos que asegurarnos que estas bien. Ya no se tarden más. Luego iré a ver qué me dicen los doctores así que no se escapen y vayan a que te revisen LEO.

LEO Y DIANA (AL UNISONO)

Ok.

DIANA ayudo a LEO a llegar a la carpa en donde se encontraba un doctor, parece que lo llamarón al instante que LEO pudo comunicarse al estar en el hoyo. El doctor reviso al muchacho y no encontró mayor novedad. Luego de ello el profesor llevo para averiguar cuáles serían los resultados que el doctor le podía decir acerca de su estudiante, al escuchar que no tendría problemas el profesor pudo relajar un poco su cara ya cansada. Le indico a LEO que estaba con permiso para no venir a trabajar toda la semana, a lo que LEO no le sentó muy bien pero al momento de apelar la resolución de su maestro, él puso una cara de enojo a lo cual el joven no pudo decir ninguna frase.

DIANA llevo a LEO hasta su casa, al momento de despedirse LEO le comenta que necesitan platicar de algo importante, a lo que la muchacha se queda algo extrañada, pero decide aceptar la invitación de su amigo y le sugiere que se vean el fin de semana, él acepta. Por último se despiden hasta el pactado día

## CAPÍTULO 2

11 DE NOVIEMBRE DEL 2012, AFUERAS DE UN CAFÉ, CENTRO DE QUITO. (10 am)

LEO llego temprano a el café en donde tendría que verse con su amiga DIANA, no estaba seguro que le iba a decir, pero sabía que tenía que decirle a alguien sobre lo que había descubierto ese día además, de otro altar para el Dios Inti.

LEO (SENTADO EN UNA SILLA AFUERA DE UN LOCAL DE CAFÉ, PENSANDO MIENTRAS TOMA UN CAFÉ FRIO)

No sé si me regañara por lo que hice, conociéndola lo hará, pero ni modo lo hecho, hecho está. Ahora a lo mejor ella me pueda ayudar a averiguar un poco más acerca del papiro. Además, de ser muy sería también es igual de curiosa que yo.

De repente ve a su amiga a lo lejos, pero parece que algo le ocurre a DIANA. Ella llega hasta la mesa en donde LEO se encontraba, saluda muy secamente a su amigo y se sienta en una de las sillas de la mesa. LEO la conoce muy bien, ella está enojada por algo, ¿será que ya averiguo lo que hizo su amigo? LEO se decide a comenzar la conversación.

LEO (MIRANDO A DIANA)

¿Qué ha pasado?, ¿Por qué estas enojada? ¿Te ha ocurrido algo?

DIANA (VIENDO A LEO)

Necesito un café bien fuerte.

LEO (MIRANDO A DIANA)

Claro, espera que llamo al camarero.

LEO estaba un poco sorprendido, a DIANA no le gustaba mucho el café, es mas no la había visto tomar uno desde hace mucho tiempo. Igualmente llama al camarero para que pueda tomar la orden de su amiga, el camarero recibe la orden y se retira de la mesa en donde estaban los dos amigos. Entonces DIANA comienza a hablar.

DIANA (VIENDO A LEO)

Ese idiota de Alberto, despidió a todos incluyendo al profesor Hurtado.

LEO (MIRANDO SORPRENDIDO A DIANA)

¿Cómo?

DIANA (VIENDO A LEO)

Si, lo que escuchaste. Parece ser que la familia de Alberto tiene muchos contactos con la Alcaldía de la ciudad y el Ministerio de Cultura, ¡lo han dejado a cargo del estudio de las ruinas! Todos se opusieron incluso el profesor Hurtado, pero eso no les hizo cambiar de opinión. Así que para quedarnos teníamos que seguir las órdenes de Alberto, cosa que nadie quiso y hecho a todos fuera del equipo de investigación. Desde ayer nadie de la Universidad, además, de Alberto, está trabajando en las ruinas.

LEO (SORPRENDIDO)

¡No puedo creerlo!, pero el profesor es alguien muy importante, ¿Por qué no le hicieron caso? ¿Y las autoridades de la universidad?

DIANA (VIENDO A LEO)

La universidad por supuesto que apoyo al profesor, pero ellos no son los responsables del hallazgo. Además, la voz del profesor fue cayada porque alguien denunció que se estaban robando piezas de las ruinas, a lo que el

profesor salió en defensa de nosotros y ahora lo están investigando. Por lo cual fue relevado de su posición y sorprendentemente su puesto fue otorgado a Alberto

JOSE (MIRANDO A DIANA)

Sorpresivamente no creo. Ya estaba todo planeado para sacarnos de ahí. Nunca me cayó ese tipo y ahora mis dudas sobre él eran ciertas. No puedo creer que el profesor también haya sido retirado de la investigación.

DIANA (MIRANDO A LEO)

Desde aquel día en que caíste al hoyo la relación entre ellos dos no fue muy buena, Alberto quería abrir hoyos por todo el templo para ver si encontraban más ruinas subterráneas, a lo que el profesor estaba en contra. Él quería que a partir del camino que encontraste seguir escavando y ver hasta donde se podía llegar y con suerte dar con la entrada a esta parte subterránea del templo.

LEO (MIRANDO A DIANA)

Ahora que tiene el control no le temblará la mano para realizar más daños al templo.

DIANA (MIRANDO A LEO)

Creo que no. Me enteré que una de las condiciones que tiene Alberto para seguir adelante del proyecto es conservar intacto el templo, es decir se seguirá el plan del Profesor pero parece que él ya no quería a alguien supervisándolo, por eso sacaron al profesor del equipo. Además, con esto quiere buscar fama y fortuna, lo que siempre ha querido.

LEO (MIRANDO A DIANA Y SONRIENDO)

Pues lo conseguirá aunque no sea su propio trabajo, con los contactos que tiene no me sorprendería verle muy pronto proclamándose el nuevo descubridor de las ruinas Incaicas. Ahora ya no me arrepiento de lo que hice.

DIANA (MIRANDO A LEO CURIOSAMENTE)

¿De qué hablas?

LEO comienza a contarle a DIANA lo que pudo encontrar en el templo subterráneo, y el objeto que se sustrajo del mismo lugar.

DIANA (SORPRENDIDA)

¡LEO!, ¡Pero qué has hecho!, si se enteran tendrán un justificativo para acusar al profesor de que se han robado algunas de las reliquias del templo.

LEO (MIRANDO A DIANA)

Tranquila. Las únicas personas que sabemos sobre el papiro somos tú y yo. Además, yo entregue el pañuelo que lo envolvía al profesor y conociéndolo seguro que lo entrego como reliquia.

DIANA (MIRANDO A LEO AUN ASUSTADA)

LEO, esto es algo muy grave. Además, va en contra de todo lo que creemos, la historia debe ser compartida con la gente.

LEO (MIRANDO A DIANA)

Lo sé DIANA, y tú me conoces, no busco fama ni nada por el estilo. Solo que este papiro me asombro mucho y fue algo más fuerte que me dijo que lo guardara y no lo mostrara a los demás. Y según lo que me contaste tome la decisión correcta, si Alberto tuviera el papiro no creo que sabría qué hacer con él. A decir verdad quiero que me ayudes a estudiarlo, aunque fue encontrado dentro del templo subterráneo, el papiro solo tienes dibujos que no me cuentan nada, no tiene lógica ni una representación simbólica clara.

DIANA (RELAJANDOSE UN POCO)

Bueno, en eso tienes razón. A decir verdad Alberto solo le interesan las ruinas y nada más. Pero eso que me dices me tiene más intrigada, ¿Cómo es eso que el papiro solo tiene dibujos sin ningún significado?, ¿Lo puedo ver?

LEO toma su mochila que traía con él y saca el papiro que guardaba en el interior de está, antes de entregárselo a DIANA se asegura que nadie les está poniendo atención, así que mira a todos lados y una vez seguro le pasa el papiro a DIANA, la cual tenía una sonrisa en la cara.

DIANA (SONRIENDO)

Jajá. Parece que estuvieras guardando un secreto de estado LEO, creo que estás un poco paranoico, no me dijiste que nadie más sabía sobre el papiro.

LEO (MIRANDO A DIANA)

Me sorprende que estés tan tranquila, además, un precavido vale por dos. Bueno en fin tómalo y dime si los dibujos significan algo para ti.

DIANA (AGARRANDO EL PAPIRO)

Veamos

DIANA con el papiro en mano lo abre y comienza a inspeccionarlo, LEO ve una leve sorpresa en la cara de su amiga y sonríe un poco. En este lapso llega la orden que pidió DIANA, toma un poco de café y continúa observando el objeto en sus manos. Pasan alrededor de 15 minutos y DIANA regresa a ver a LEO.

DIANA (MIRANDO A LEO)

Los dibujos son conocidos, son representaciones de los dos Dioses Mayores Incaicos, el Sol y la Luna. Se nota que la diosa reverencia al Sol, al estar en una posición alrededor con respecto a esté. Lo que me atrajo más la atención fueron las palabras debajo de cada uno de estos, tienen nombres de Iglesias.

LEO (MIRANDO A DIANA)

¿De las Iglesias?, ¡es verdad!, se me había pasado por alto ese dato.

Efectivamente cada dibujo tenía un nombre debajo de su representación, El Dios Sol tenía esto escrito “Santo Domingo, un nombre de mucha fuerza en estos tiempos, busca la misma fuerza y poder del primer Inca en sus dominios”, La Diosa Luna tenía escrito “En San Francisco encontrarás la sabiduría de la primera reina del Imperio del Sol”. LEO una vez más se daba cuenta de la inteligencia de su amiga, él pensaba que eran los nombres de unos santos pero no le encontraba significado alguno. Pero ¿Qué más podían significar estos nombres?

LEO (MIRANDO A DIANA)

¡Eres increíble!, te has dado cuenta en un solo instante, pero aún no lo entiendo, ¿Qué más puede significar estos dibujos?

DIANA (MIRANDO A LEO)

Para mí es un mapa. La forma en que están dibujados me recuerda a la posición de las iglesias en un mapa. Déjame tomarle una foto y la sobreponemos al mapa de Quito y veamos si mi hipótesis es correcta.

LEO asiente y enseguida DIANA saca su teléfono celular de su bolso al igual que una computadora. DIANA toma la foto del papiro y la descarga a su computadora. Se dispone a sobreponerla en el mapa de Quito y efectivamente las posiciones concuerdan con las de las iglesias dibujadas en el mapa. DIANA sonríe.

DIANA (SONRIENDO Y MIRANDO A LEO)

Mira LEO, como te lo dije. Las iglesias y los dibujos concuerdan con su posición. Creo que lo que tenemos en mano es una especie de mapa, y las pistas que dan deben estar relacionadas con algo dentro de las iglesias.

LEO (MIRANDO LA COMPUTADORA)

Se conoce que mucho del arte de Quito tiene una mezcla entre las creencias cristianas e incaicas, a lo mejor dentro de ellas debe haber conexiones con estas pistas, pero ¿Cómo se puede averiguar? Si se lo decimos a alguien debemos mostrarles el papiro y el profesor estará en problemas.

DIANA (SONRIENDO)

No necesariamente. Estamos a pocos días de las fiestas de Fundación de Quito, el 6 de Diciembre, en cada año los turistas como la demás gente puede visitar las iglesias y monasterios insignes de la ciudad sin ninguna restricción. Y muchas veces se realizan trabajos de limpieza y reconstrucción antes de esas fechas, si hablamos con el profesor de seguro que él pueda asegurar un puesto de limpieza en estas iglesias y podremos acceder a ellas con total libertad.

LEO (MIRANDO A SU AMIGA)

¡Eres una maravilla!, Alberto no sabe lo que perdió cuando te despidió.

DIANA (MIRANDO SERÍA A LEO)

Espera LEO. Antes que nada quiero que me prometas que lo que lleguemos a encontrar será puesto a conocimiento público.

LEO (ASINTIENDO)

Claro que si DIANA, no te preocupes. Tenía la misma idea en mi cabeza.

DIANA (SONRIENDO A LEO)

Entonces cuenta conmigo.



Dicho esto los dos jóvenes se quedaron planeando un poco más sus próximos movimientos y como explicarían el súbito interés de ir a limpiar las iglesias para así poder expresarlo a su maestro.

### **CAPÍTULO 3**

#### **SAN FRANCISCO**

\*El primer nivel del juego se realizará en esta iglesia, se debe presentar mediante una secuencia de video a la iglesia y como lograron ingresar al cuerpo de restauración los jóvenes. Lo más recomendable es hacerlo mediante caricaturas.

Después de meditarlo mucho, los jóvenes decidieron contarle a su profesor acerca de lo que habían encontrado y a diferencia de lo que pensaban, su maestro les brindo un total apoyo a su pequeña expedición, incluso les aseguro que podría ayudarles si en algún momento necesitaban de su conocimiento. Los chicos estuvieron encantados con la idea.

Así sin más, y como lo había pronosticado DIANA, en las iglesias de la ciudad de Quito, se estaban realizando trabajos de limpieza y restauración de algunas partes de ellas como de imágenes propias de cada una de los templos cristianos. Su profesor logró contactar con el jefe de los trabajos en los templos, Julio Aguirre, un hombre de algo más de 50 años y un gran amigo de él. Pero para mala suerte de Diana, solo pudo conseguir un puesto para LEO dentro de los trabajos de restauración.

13 DE NOVIEMBRE 2012, 12pm (En la iglesia de San Francisco)

Una vez ingresado al cuerpo de restauración de las iglesias, la primera parte del plan para investigarlas estaba realizado, ahora tenían que encontrar una manera en que Leo pueda estar solo dentro del recinto religioso para poder revisarlo, esto solo era posible a la hora en que los trabajadores tomaban su almuerzo, esto le daba más o menos una hora para poder investigar el recinto.

LEO (HABLANDO A DIANA A TRAVÉS DE UNA RADIO)

Bueno ahora tenemos 1 hora a lo mucho para revisar la iglesia.

DIANA

¡Si, apresúrate!, tal vez no tengamos otra oportunidad como ésta

\*Dicho esto, se debe pasar a modo jugador, es decir tenemos que pasar el control al jugador para que pueda investigar la iglesia.

Las animaciones en este recinto son las siguientes:

Al revisar los pilares dentro de la iglesia, se debe notar el símbolo de la luna Inca.

También se debe hacer una animación que muestre la misma imagen de la luna Inca en un ventanal de la iglesia, el cual debe ser destruido para poder dar paso a un rayo de luz.

El rayo de luz debe impactar a una imagen de la luna en un pilar, este debe brillar y luego proyectar esa luz a otro pilar, y así en tres pilares diferentes dejando una marca de la luna inca.

En el último pilar debe reflejar la luz al suelo y dar paso a una puerta secreta.

Esta puerta debe de abrirse en una animación para dar paso al juego propio Inca.

\* Una vez descubierta el pasaje al juego Inca el usuario debe acceder a éste.

Juego Inca

El juego a desarrollar en la primera iglesia se trata acerca de la creación del mundo a través de los ojos incaicos.

En las creencias Incas, existieron dos creaciones. En la primera creación, el Dios Viracocha caminaba por las inmensas y desiertas pampas de la meseta Viracocha

Pachayachachi, "el hacedor de las cosas", después de haber creado el mundo en un primer ensayo como si se tratara de un bosquejo, sin luz, sin sol y sin estrellas. "Pero cuando vio que los gigantes eran muchos más grandes que él, dijo: No es conveniente crear seres de tales dimensiones, ¡me parece mejor que tengan mi propia estatura! Así creó Viracocha los hombres según sus propias medidas, tal como son hoy en día, pero aquellos vivían en oscuridad".

Viracocha ordenó a los hombres vivir en paz, en orden y como honrarle, pero aquellos se entregaron a la mala vida, los excesos y fue así como el **Dios** creador, los maldijo, convirtiéndolos en piedras o animales, algunos quedaron sembrados en la tierra, otros fueron absorbidos por las aguas, finalmente arrojó sobre ellos un diluvio en el cual todos perecieron.

Esta historia se contará a través de dibujos, tenemos que mover las piezas de esos dibujos que representen a:

Dios Viracocha

El mundo

Los gigantes (primeros hombres)

Conversión del tamaño de los hombres (reducción de tamaño)

Viracocha ordenando a los hombres vivir en paz

Los hombres cayendo en la maldad

Viracocha castigando a los hombres

Para ponerlos en orden, que concuerde con la historia antes descrita.

Al resolver el juego se mostrará una secuencia de video que dejará ver la primera parte de una llave Inca, dividida en dos partes.

LEO (riendo)

¡Muy bien!, hemos conseguido este extraño artefacto, parece la parte superior de una llave, creo que encontraremos la otra parte en la siguiente iglesia.

DIANA (a través de la radio)

¡Es lo más seguro! Pero Leo toma fotos del mecanismo de este panel, no parece ser de una civilización atrasada como se pensaba de los Incas.

LEO (mirando el muro)

¡Tienes razón!, tal vez encontraremos algo más que vajillas o tesoros dentro de nuestra aventura

DIANA (a través del radio)

Vamos Leo, ya casi es la hora en que regresan los demás

LEO

Si

Una vez que ha salido del pasillo subterráneo donde estaba el juego incaico la puerta secreta debe cerrarse sin dejar rastro, Leo muestra las imágenes incas a los trabajadores y se quedaran asombrados lo cual generará que comiencen a estudiarlas.

## **CAPÍTULO 4**

SANTO DOMINGO

14 Noviembre del 2012

Dado los descubrimientos en la iglesia de San Francisco, las miradas se habían dirigido a ese recinto, esto dio un poco más de facilidad el que Leo pueda revisar la iglesia de Santo Domingo, la segunda mencionada en el mapa, casi todos los trabajadores que estaban en este recinto se habían dirigido a la iglesia de San Francisco por lo cual en, Santo Domingo necesitaban manos y ahí funciono una vez más los contactos del Profesor.

Por suerte Leo fue designado al interior de esta iglesia, aquí solo una persona, además, de él, fueron encargados de realizar trabajos de reconstrucción en esta iglesia, al igual que en San Francisco, en Santo Domingo esperaron a que su compañero de trabajo saliera a almorzar para poder investigar el recinto, una vez llegada la hora comenzó la investigación de la iglesia.

LEO (MIRANDO EL MAPA)

Según el mapa, dice que aquí encontraremos la fuerza de Ayar Manco, que fue el primer inca.

DIANA (RIENDOSE A TRAVÉS DEL RADIO)

Eres muy generalista, a lo mejor encontremos algo referente a Ayar Manco no me imagino cómo podremos encontrar algo como su fuerza jajá.

LEO (MIRANDO LA RADIO)

Chistosa

\* Aquí se debe cambiar al modo jugador, al igual que en la primera iglesia se debe investigar la iglesia, esta vez será un poco más fácil encontrar las pistas para averiguar la ubicación de la puerta secreta al juego incaico.

En una parte del piso de la iglesia, se debe mostrar un dibujo incaico, esté marcará el switch o botón que accionará la animación para mostrar unos pilares que deben levantarse del piso.

En la base de los pilares, se representarán imágenes incas del Sol, recordando que Ayar Manco era el hijo del Dios Sol, es muy viable representarlo como un Sol diferente al que tiene su padre, esto se debe diseñar porque no se ha encontrado una imagen representativa de Ayar Manco.

El juego aquí consistirá topar los pilares en un orden establecido por las imágenes, éstas deben representar al sol en una periodo de tiempo, algo parecido a la representación de la luna en los calendarios, es decir como luna nueva o luna llena (con una parte de esta oscura y la otra clara), solo que aquí utilizaremos la imagen del sol representando a Ayar Manco, los pilares al ser tocados descenderán a su posición inicial, si el juego no es resuelto adecuadamente, el botón debe sobresalir para comenzar el mini-juego nuevamente, caso contrario se deberá mostrar unas escaleras que estén ocultas en el piso de la iglesia.

En el siguiente juego Incaico, se contará la continuación al primero es decir la siguiente historia:

Solo tres hombres quedaron con vida, y con el recado de ayudarlo en su nueva creación, luego de pasado el diluvio, "el maestro del mundo", decidió dotar a la tierra con luz, y fue así como junto con sus tres súbditos ordenó que brillase el sol, la luna y las estrellas y ocuparan su lugar en el vasto firmamento.

Aquí el objetivo del juego será proveer de luz a los símbolos del sol, la luna y las estrellas, al primero ya cuenta con una luz, el objetivo será diseñar un sistema de espejos para ponerlos de tal manera que puedan reflejar una luz de una linterna y que pueda impactar a los dos símbolos restantes. Al realizar esto se obtendrá la segunda parte de la llave.

Al obtener la segunda parte de la llave, Leo sufrió una especie de visión, en la cual se le mostro la imagen de una montaña, Los Llanganates, con lo cual marcó su siguiente destino.

## CAPÍTULO 5

Gracias al último fragmento de la llave, Leo pudo ver los montes Llanganates en una visión que lo condujo a los mismos. Esperando encontrar algo que pudiera abrir la llave en su poder.

LEO

Estoy en los Llanganates, pero según mi visión la parte Oeste de los mismos me conducirán a la respuesta que estamos buscando, el problema es que no veo una ruta o camino que pueda seguir, lo más seguro es que tenga que escalar la montaña.

DIANA (POR LA RADIO)

Por suerte yo tenía ese kit de escalada, espero que te sea de ayuda.

LEO

Eso tenlo por seguro.

\*Aquí tendremos que diseñar la siguiente parte del mini-juego, uno de escalada, esto será diseñado si tenemos el tiempo suficiente para hacerlo, caso contrario solo tendremos que poner una representación de un escalador como imagen estática.

Una vez en la cima, la llave comenzará a brillar y aparecerá un templo inca escondido por la maleza, la llave será atraída al interior del mismo y Leo tendrá que seguirla, una vez dentro el protagonista tendrá una nueva visión en la cual tendremos que contar la siguiente historia.

Rumiñahui, hablará directamente a la mente del protagonista y detallará la llegada de los españoles, la traición de algunos incas y que este fue el principal motivo de perder la batalla contra los invasores.

Dejando al final la verdadera riqueza que lega Atahualpa a través de su general, el de que un pueblo siempre debe estar unido en los momentos difíciles y si existe alguna traición, está puede ser tan grave como para destruir todo un imperio, todo un porvenir, toda una nación.

Esto es preferible hacerlo mediante imágenes estáticas con leyendas sobre ellas que puedan contar la historia. De esta manera, el jugador podrá entender mejor el mensaje del juego.



## Anexo 2. Focus Group

### Objetivo

Evaluar la funcionalidad del juego para dar a conocer la historia del Reino de Quito, específicamente del tesoro de Atahualpa de una manera didáctica y entretenida.

### Metodología

A través de un Focus Group realizado a niños entre 8 y 12 años de edad.

### Resultados

Después de realizar un Focus Group con 4 niños y 1 niña, con edades que oscilan entre los 9 y 12 años llegamos a las siguientes conclusiones:

Preguntas	Pablo Gallegos	Santiago Jaya	Mateo Balarezo	Paula Balarezo	Emilio Gonzales
¿Te gustan los videojuegos?	sí	sí	sí	sí	sí
¿Cuál fue tu primer videojuego?	Mario Bros	DragonBall	StarWars lego	Mario Bros	Donkey Kong
¿Tienes alguna consola?	Xbox	Play Station 2	Play Station 3	no tiene	Play Station 3
¿Antes de jugar tenías conocimiento sobre el tesoro de Atahualpa?	no	no	sí	no	sí
¿Aprendiste algo de nuestra historia después de jugar?	sí	sí	sí	sí	sí
¿Qué es lo que te gusto del videojuego?	las herramientas	el movimiento	las montañas	los niveles	las montañas
¿Consideras que los niveles son difíciles de pasar?	no	sí	sí	sí	no

¿Si es un sí, que niveles fueron difíciles?	-	el nivel 2	El nivel 1	el nivel 2	-
¿Lo volverías a jugar?	sí	sí	sí	sí	sí

### **Apreciación personal sobre las preguntas realizadas**

Su primer videojuego fue Mario Bros, Dragon Ball, Donkey Kong y Star Wars Lego, todos cuentan con consolas como Xbox, Play Station 2, Play Station 3 y Wii, antes de jugar de los 5 niños solo 1 tenía conocimiento de la historia del tesoro de Atahualpa, todos aprendieron del videojuego, de los 5 niños solo a 2 niños les pareció difícil pasar los niveles y específicamente les pareció difícil pasar el nivel 1, a todos les gusto las montañas del videojuego y el movimiento del jugador entre los lugares que recorre, todos los volverían a jugar.

### **Factores motivantes del videojuego.**

- Enfocado a niños y jóvenes
- Qué es un videojuego educativo y didáctico para la enseñanza de la historia del Ecuador
- La historia atrae y no se necesita conocer para jugarlo
- Es un videojuego corto y ligero que ayuda a resolver los niveles con facilidad

### **Factores limitantes del videojuego.**

- No existe un instructivo
- Al principio no se sabía bien cual eran los movimientos que se debía hacer con el mouse
- La resolución de la pantalla debe ser correcta para que funcione adecuadamente el juego

Después de realizar pruebas se determinó que los niños presentan diferentes reacciones tales como: angustia, emoción, frustración al momento de no saber cómo pasar un nivel y satisfacción al pasar los niveles o ganar, algo que también se tomó a consideración es que no se notó ninguna alteración de los niños para renunciar al reto que les presentaba el juego.

Se les propuso el juego con el fin de observar las reacciones como usuarios; para ver si entendían la historia de Atahualpa y que mensaje les dejaba. Jugaron por aproximadamente 30 minutos, las primeras reacciones del grupo fueron de interés y aceptación, en los primeros minutos exploraron el primer nivel y verificaron como se movía el personaje en primera persona por el escenario, verificaron y preguntaron si existían ayudas para pasar el juego. No se necesitó una mayor intervención externa con esto se identificó que el juego no requiere de instrucciones verbales, además, se pudo notar que existe un nivel de complejidad y que los usuarios superan los retos con un nivel relativo de facilidad.

Se observó que los niños de entre 10 y 12 años tienen más interés en saber cómo se realizó el juego y como fue programado, además, se informó que es necesario que se ponga un poco más de retos en los diferentes niveles. El tiempo de control no fue suficiente para determinar cuánto se puede demorar los usuarios en pasar completamente el juego que consta de 4 niveles ya que algunos niños no lograron pasar del segundo nivel porque no entendían de buena manera la historia del primer inca, y esto motivo a que pregunten cual era esa historia.

Otro caso que fue de gran interés es el de la relación existente en los pilares del primer nivel, ya que la mayoría de niños no conocían que tanto la luna y el sol tenían gran importancia en la historia de los incas, se nota que para que el juego sea entendido de una mejor manera es necesario que un maestro que sepa de historia o que tenga conocimientos de esto se encuentre junto a ellos para que explique un poco sobre leyendas; con esto se logró que los niños despierten gran interés en la Historia del Ecuador, en especial de la historia de Atahualpa.

Se notó también que los maestros que no pertenecían al grupo experimental, también tenían curiosidad hacia el juego y se interesaron en realizar evaluaciones.